FLORE
Repository istituzionale dell'Università degli Studi di Firenze

MyStoryPlayer: experiencing multiple audiovisual content for education and training

Questa è la Versione finale referata (Post print/Accepted manuscript) della seguente pubblicazione:

Original Citation:

MyStoryPlayer: experiencing multiple audiovisual content for education and training / P. Bellini;P. Nesi;M. Serena. - In: MULTIMEDIA TOOLS AND APPLICATIONS. - ISSN 1380-7501. - STAMPA. - (2014), pp. 1-41. [10.1007/s11042-014-2052-9]

Availability:

This version is available at: 2158/956910 since:

Published version:

DOI: 10.1007/s11042-014-2052-9

Terms of use:

Open Access

La pubblicazione è resa disponibile sotto le norme e i termini della licenza di deposito, secondo quanto stabilito dalla Policy per l'accesso aperto dell'Università degli Studi di Firenze (https://www.sba.unifi.it/upload/policy-oa-2016-1.pdf)

Publisher copyright claim:

(Article begins on next page)

05 May 2023

Multimed Tools Appl
DOI 10.1007/s11042-014-2052-9

MyStoryPlayer: experiencing multiple audiovisual content for education and training

Pierfrancesco Bellini · Paolo Nesi · Marco Serena

Received: 1 November 2013 / Revised: 24 February 2014 / Accepted: 24 April 2014
© The Author(s) 2014. This article is published with open access at Springerlink.com

Abstract There are several education and training cases where multi-camera view is a traditional way to work: performing arts and news, medical surgical actions, sport actions, instruments playing, speech training, etc. In most cases, users need to interact with multi camera and multi audiovisual to create among audiovisual segments their own relations and annotations with the purpose of: comparing actions, gesture and posture; explaining actions; providing alternatives, etc. Most of the present solutions are based on custom players and/or specific applications which force to create custom streams from server side, thus leading to restrictions on the user activity as to establishing dynamically additional relations. Web based solutions would be more appreciated and are complex to be realized for the problems related to the video desynchronization. In this paper, MyStoryPlayer/ECLAP solution is presented. The major contributions to the state of the art are related to: (i) the semantic model to formalize the relationships and play among audiovisual determining synchronizations, (ii) the model and modality to save and share user experiences in navigating among lessons including several related and connected audiovisual, (iii) the design and development of algorithm to shorten the production of relationships among media, (iv) the design and development of the whole system including its user interaction model, and (v) the solution and algorithm to keep the desynchronizations limited among media in the event of low network bandwidth. The proposed solution has been developed for and it is in use within ECLAP (European Collected Library of Performing Arts) for accessing and commenting performing arts training content. The paper also reports validation results about performance assessment and tuning, and about the usage of tools on ECLAP services. In ECLAP, users may navigate in the audiovisual relationships, thus creating and sharing experience paths. The resulting solution includes a

P. Bellini · P. Nesi (✉) · M. Serena
Distributed Systems and Internet Technology Lab, Department of Information Engineering, University of Florence, Via S. Marta 3, Florence 50139, Italy
e-mail: paolo.nesi@unifi.it
URL: http://www.disit.dinfo.unifi.it

P. Bellini
e-mail: pierfrancesco.bellini@unifi.it
URL: http://www.disit.dinfo.unifi.it

M. Serena
e-mail: marco.serena@unifi.it
URL: http://www.disit.dinfo.unifi.it

Published online: 02 August 2014

uniform semantic model, a corresponding semantic database for the knowledge, a distribution server for semantic knowledge and media, and the MyStoryPlayer Client for web applications.

Keywords Audiovisual relations · Educational tools · Annotation tool · Multi video synchronization

1 Introduction

The way media usage for entertainment and edutainment is rapidly undergoing a range of transformations. New formats and content fruition modalities are appearing. Also on television, new ways of interactions with users have been proposed, through multistream TV programs especially with the introduction of web and IPTV, synchronized multiples views/streams of the same event, and multiscreen/device experiences. In Internet, social networks and web sites are distributing videos (such as YouTube, Vimeo, etc.), providing support for nonlinear play of audio visual: augmenting the content under play with links and connections with other content, e.g., providing suggestions. Web based solutions provide links to change context and thus a virtually unlimited number of combinations/paths. This fact is very attractive to users for both entertainment and learning, *since it allows the creation of personal stories and experiences*. Therefore, many web social media and web based applications are proposed, where the user can synchronously play video and slides, can see a video and its related chatting channel, can see a video and jump to another audiovisual, can share content (audio visual, comments, votes, tags, etc.). Moreover, social networks are widely used by many educational institutions as content delivering networks facilities, even if they do not cope with pedagogical and didactical points of views, due to their lack of annotation structure, definition of relationships, formal model for classification, aggregation, composition, etc. Indeed, with the popularity and the huge growth of audio visual data on internet and the omnipresence of large-scale multimedia database, any efficient access to them is becoming more and more essential as a first step to provide educational tools.

In this paper, MyStoryPlayer solution is presented. MyStoryPlayer has been designed to cover a set of scenarios and requirements related to education and training described in the next sections.

1.1 The MyStoryPlayer scenario and rationales

There are several edutainment fields where multi-camera shooting is a traditional way to grab and report events and activities. This happens in: (i) performing arts and news (taking different points of view of the same scene, recording workshops and master classes from different points of views); (ii) medical imaging for training, where the surgical action is taken from inside endoscopic camera, from the outside to show the action of the surgical team, and plus other views on monitoring instruments; (iii) sport events (offering multiple views, and giving the user the possibility to select them), and in (iv) some TV programmes such as the big brother.

Moreover, there are many other applications where there is the need of aligning/relating different audio visual segments to play them together so as to compare the scenes, even if they have not been taken from the same event. This kind of activity is performed for enriching details, adding comparative examples and comments, providing alternatives. Examples are in the: (a) performing arts and films analysis (comparing and/or highlighting different postures and actor gestures, different performance of the same opera, director and scenery Citations, alternatives scenes, etc.), (b) music education and training (comparison with the teacher, with previous performances, providing correction), (c) sport education and training (comparison

with competitors, against correct postures and gestures; comparing different performances), (d) medical and surgical training, (e) public speech training (showing different points of view of both the same event and human behaviour), etc.

Synchronization and comparison among audiovisual media segments can be produced by creating media relationships and then annotations, which can be regarded in the educational perspective as a constructivist interaction and experience. A generalization of these aspects also includes the annotation of audio-visual segments with other shorter/longer ones (comparison, correction, etc.), the association of an image to a video segment or instant to be shown for a while, the possibility of jumping from a related audio visual to another (with the corresponding change of context, which should show the annotations of the latter video and not the ones related to the former video) and possibly the chance of jumping back to the previous video with a button as it occurs with every browser; the possibility of recording the sequences of such actions to share them among students and colleagues, etc. A large set of possible experiences and their combinations can be produced by exploiting a limited number of common audiovisual, similarly to the user navigation on the set of html pages on the web.

The rendering of multiple video streams synchronized together presents complexities, if you would like to avoid delay in the change of context. For example, this occurs when passing from a multicamera view where four videos are synchronously in execution, to a different set by clicking on a proposed annotation. Moreover, the wide range of user interactions to establish relationships among audio-visual, the navigation among audiovisual (change of context), the experience recording, the experience playback and the needs of precise synchronizations make the design and implementation of a complete solution complex. For the same reason, these latter requirements lead to exclude the possibility of realizing the solution by using a preassembled streaming or the adoption of simple web pages containing synchronized playable audiovisual produced HTML.

1.2 Contribution of this article

According to the above presented scenarios, a web based solution (models and tools) to create relationships/synchronizations, to navigate among audiovisual media, and to save and share user navigations is needed. Some of the relationships among the audiovisual content may be established since the very shooting phase of some events, or later on during a qualified authoring phase by experts. On the other hand, final users, as students and researchers, need to provide their own contributions, while performing their own analyses, providing comments, creating annotations, and thus producing and sharing their experiences and relationships. This is part of the social media learning solutions, which is at present expected by final users.

This paper focus is on the MyStoryPlayer/ECLAP model and solution. The major contributions of this article to the improvement of the state of the art are as follows:

- The annotation models at the state of the art are not satisfactory to model all the aspects needed by the MyStoryPlayer as highlighted in the rest of the paper. Therefore, the definition of MyStoryPlayer/ECLAP annotation model has been performed. It is an RDF based annotation [44] model to formalize the relationships among audiovisual. The MyStoryPlayer model solution is grounded on temporal patterns which are conformant with the concepts adopted in the education and training scenarios. The proposed RDF based annotation constructs can be progressively obtained by a client player to change the audiovisual context in real time, including synchronized and related audiovisual, thus avoiding the reconstruction of the whole web page. A part of the proposed MyStoryPlayer/ECLAP annotation model is exported as Linked Open Data using Open Annotation data

model developed by W3C. The MyStoryPlayer/ECLAP annotation model is part of the ECLAP ontological model described in [4], to specifically manage the audiovisual annotations.

- A user experience can be regarded as possible trace on both the meshes of annotations and user's actions performed during the playing of synchronized and annotated audiovisual (clicks and context change at specific time instants). The state of the art in this case is totally absent, therefore what has been carried out is the definition of a formal trace to save and share experiences on playing and navigating among synchronized and annotated audiovisual. The produced experiences can be saved and re-proposed to other users, thus replicating the user experience without creating a specific set of annotations on the same content of other annotations and audiovisual.
- Design and development of algorithms and tools to shorten the production of audiovisual relationships/annotations, to mine the audiovisual received metadata with the aim of suggesting the possible relationships among audiovisual to the expert users during the production and authoring of the relationships. The tools have been collaboratively used within the ECLAP community to create a set of relationships among audiovisual which are currently used for education and training.
- Design and development of the MyStoryPlayer solution. It includes (i) a server working for the ECLAP Social Network to produce, collect and distribute related and annotated audiovisual content; and a service to record and share user experiences , and to search annotations and relationships; and (ii) a client tool, called MyStoryPlayer Client, which provides support for navigating among the related and annotated audiovisual by starting from a reference content. MyStoryPlayer allows the user to browse and analyse the several relationships among audiovisual: clicking on the annotations, changing the context and producing his/her own personal experience among played segments of the proposed network of relationships. The personal constructed experiences can be saved and shared with other users. MyStoryPlayer is one of the annotation tools promoted by Europeana, European digital library of cultural heritage content [http://pro.europeana.eu/web/guest/thoughtlab/enriching-metadata].
- Provide a solution to improve the quality for synchronous rendering of audiovisual content in presence of low bandwidth conditions. In such cases, if several videos are played on the same web page progressively downloaded via http protocol they are typically affected by large diverging delays. The synchronization problems are even higher in presence of: direct jump backward and forward with respect to the play execution time; swap from one video to another (i.e., master and annotation); back trace along the stack of performed swaps. The aim is to keep the desynchronization limited. This is much more critical in presence of low bandwidth and long video duration where relevant delays need to be corrected.

The MyStoryPlayer/ECLAP has been developed for supporting ECLAP social learning on a best practice network environment, the performing art content aggregator of Europeana. ECLAP is the European Collected Library of Performing arts (http://www.eclap.eu), a collaborative environment to produce enriched content and metadata for content collections that are posted on Europeana in terms of EDM and made accessible as LOD (Linked Open Data) [22]. ECLAP is a network of 35 prestigious performing arts institutions, and it is used for education and research goals. ECLAP partners provide content in 13 different languages, the Consortium's geographical areas being mainly in Central Europe, plus Chile, South Africa, Russia. ECLAP has about 170,000 content elements, ranging from video, audio, documents,

3D, images, braille, etc., including performance, premier, libretti, scores, pictures, posters, manual designs, sketches, etc.

This article is organized as follows. Section 2 reports an overview of the related work to highlight the state of the art, with major problems depending on the proposed systems and solutions. The analysis stresses the similarities, differences, and problems to cope with the above mentioned scenarios. In Section 3, the MyStoryPlayer model of relationships among media is presented by showing the formalization of relationships with respect to the mentioned requirements and scenarios. In Section 4, the MyStoryPlayer is shown during its usage by providing a comprehensive example of both navigation among the relationships and how the user experience is recorded. Section 5 shows the semantic model of MyStoryPlayer and an example of the model managed by the client tool during the play of multiple media, and its export in terms of LOD. Section 6 shows how the media relations are automatically produced and some details about manual editing. In Section 7, the MyStoryPlayer architecture is described and some details have been added on how the synchronization problems were solved. To this end, outcomes regarding the assessment on the obtained results are also proposed. Section 8 provides some usage data about the adoption of MyStoryPlayer by the ECLAP users. Conclusions are drawn in Section 9.

2 Related work

This section reviews the state-of-the-art techniques on audiovisual modelling and fruition for entertainment and edutainment. In particular, it focuses on three aspects: (i) content modelling for synchronized and interactive audiovisual rendering via web, (ii) media annotation modelling and tools, and (iii) authoring audiovisual annotations for streaming and web applications.

2.1 Content modelling for synchronized audiovisual rendering via web

There is a large range of content formats and languages that can be used to model cross media content with synchronizations and relationships, such as standard: MPEG-21 [18], MXF [1] AXMEDIS/MPEG-21 [3, 5], SCORM/IMS [46], MPEG-4 [43], and proprietary formats as Adobe Flash, MS Silverlight, etc. In [12], media synchronization models are analysed taking into account a large range of former models and formats. In most cases, the presentation layer of these formats are formalized by using specific description languages, such as SMIL [17], HTML/HTML5, MPEG-4 BIFS [43], or the less known models and languages as ZYX [13], NCL [26]. Among these formats, we should focus on the ones that can be managed to set up a *web based collaborative environment* where several users can access synchronized and annotated audiovisual to manipulate and play interactively the models. Among them, SMIL has been designed to produce interactive presentations, and it may have links to other SMIL presentations and graphical elements to allow the user interaction. SMIL provides support to model transitions, animations, synchronization, etc. In SMIL 3.0, variables can be added to the multimedia presentation, thus enabling adaptation to user interactions. Despite its expressivity, SMIL is still a page description language which needs to reload the page to change context and it is at present not very well supported by web based players and browsers. This means that you have to install a plugin into the browser in order to exploit all SMIL capabilities and constructs, then you have to modify the plugin code to cope with the progressive access to the information related to the audiovisual relationships and interactivity details of the client user page. SMIL is mainly typically supported by non-web players such as Ambulant tool. On the other hand, in [23], a SMIL interpreter able to render a reduced set of SMIL constructs on

HTML pages has been created by using javascript. This approach is still unsuitable for rendering and controlling synchronized videos. SMIL and HTML have been compared and used in AXMEDIS/MPEG-21 for the presentation layer modelling of AXMEDIS content, exploiting the SMIL Ambulant player library [8]. Alternative solutions could be grounded on the adoption of HTML5 with the usage of JavaScript, or the usage of Adobe Flash or MS Silverlight. Such solutions may exploit internal features to model and describe the graphic layout of the web page, and interactive aspects, while some script is needed to cope with the access to the progressive information related to the change of context, jump to a different media context, add an annotation, etc.

A relevant problem refers to the difficulties of creating a web based playing of multiple synchronized videos. In this sense, HTML5 as well as Flash or Silverlight do not provide direct solutions. In more detail, a time skew and delay in the synchronizations can be detected among videos when multiple video streams are played / started on the same web page. Typically, the time skew is due to latency and variance in seeking the http based video access and stream, i.e., de-synchronization [31]. These problems may depend also on the network connection reliability. Moreover, when the user passes from the execution of a multiple video streams on a web page to another set of videos that should start at a selected time instant (i.e., the user changes context), specific techniques to seek the streams from a different starting point of the video are needed.

A different approach could be based on the production of an integrated stream combining multiple streams and maintaining the inter-stream synchronizations, for example by using a unified muxed transport stream [35]. A comparative study on inter-stream synchronisation has been proposed in [14], while performance aspects have been reviewed in [55]. Optimization and correction mechanisms for video streaming have been proposed in [19, 32, 53]. This latter has adopted a scalable and adaptive video streaming model. Multiple video streams could be provided by using RTSP server to some custom players to be kept synchronized according to the RTSP protocol which constraints to establish specific client–server connection for each stream. On the other hand, the inter-stream synchronisation approach does not provide the needed flexibility and the interactivity of web based solutions that may allow swapping from many connected relationships among a network of annotations, among media located in different servers, without the direct control of the streaming server. Moreover, seeking, preload and discharging techniques can be applied on specific file formats and keep under control the client player of the context. In [36], a solution for intelligence download and cache management of interactive non-linear video has been presented. The *proposed solution* aimed at minimizing the interruption when the nonlinear video changes context due to the user interaction. The adopted approach was based on exploiting the knowledge of the structure of the nonlinear video and the adoption of a prefetch approach.

In [24], an analysis has been performed to solve the mentioned problems by using a combination of prefetching (preloading and cache) and frame discarding mechanisms, thus solving the problem by increasing accuracy and reducing delay. Experiments have been performed by means of a graphic page built in SMIL. Therefore, the optimization constrained to modify the Ambulant player for SMIL 3.0, and not a common web browser. Additional problems have been detected related to the: (i) density of the seeking points in different video formats, (ii) highly compressed video file formats for which the seeking points are not regularly placed, (iii) saturation of the available connection bandwidth with respect to the frame rate and compression of the videos. The solution provided in [24] adopted raw videos to keep the delay limited. This solution is unsuitable for web page tools and has strong limitations related with the impossibility of using compressed videos.

As a concluding remark, the adoption of HTML5 or flash, as well as the usage of SMIL in customized Ambulant-based player as in AXMEDIS and in [24] cannot be regarded as the solution of the above identified problems. These formats are powerful coding tools for modelling the graphic page and the user interaction (including the accommodation of multiple videos on the same page). Yet, they are unsuitable to solve major problems, such as: how to model the audiovisual relationships/synchronizations and pass them to the client side without reconstructing the whole web page; how to progressively access to the needed information only at each change of context; how to cope with saving and reloading user experiences; and how to make the change of context (jump, swap and back) smooth and clean (keeping synchronizations among videos), with the aim of providing a higher quality of experience for the final user.

2.2 Media annotation modelling

The literature presents a large number of multimedia (audio visual) annotations models and tools including both the modalities to annotate audiovisual and play them (i.e., play the media and see, contribute with other annotations). Most of them are grounded on semantic descriptors modelled as media or multimedia annotations formalized in MPEG-7 [39] and/or recently in RDF as in Open Annotation of W3C [41]. Moreover, a number of other solutions should be mentioned. Vannotea solution has been proposed for collaborative annotation of videos [30, 45]. The main ideas of Vannotea are meant to allow the collaborative discussion on video content in real-time, thus producing annotations as comments formalized as MPEG-7 and Dublin Core [21]. The DC is also used to index, search and retrieve video and single segments. Annotations are not typically used to establish relationships among videos (audiovisual) but only to add on audiovisual segments some text, descriptors, etc. In that case, the annotation database has been developed by exploiting the Annotea model and solution. Annotea is a solution proposed by the Semantic Web Advanced Development group of W3C [27, 29]. The idea of Annotea is to create annotations which can refer to the annotated media. Annotations are modelled as RDF. Xpointer is adopted to link them to the annotated media content. RDF-based annotations can be searched by using semantic queries, for example in SPARQL. Single annotations are associated with html/xml elements on the web. For their execution and connection to the media source (i.e., play and rendering) an *extended version* of Mozilla Firefox has been provided, thus constraining the user to adopt a specific browser. In [8], AXMEDIS/MPEG-21 [6] has been exploited to create annotations in cross media content. These cross media annotations are overlapped to images, audio and videos, and may refer to single elements/essences included into cross media content by using AXMEDIS internal links and protocol. AXMEDIS annotations may be produced, collected, and shared and synchronisation could be managed as well via MPEG-4 standard, whereas MPEG-21 had been initially formalized as a set of descriptors for content modelling and distribution [38].

There is an ongoing work in the W3C Open Annotation Community Group for the development of a common data model and ontology for the representation of annotations [41], a working draft was produced in Feb. 2013 http://www.openannotation.org/spec/core/. This group was jointly founded by the Annotation Ontology and the Open Annotation Collaboration. The OpenAnnotation model is based on one or more ‘Target’ elements referring to the digital resources (or its part) being annotated and some ‘Body’ elements representing the body of the annotation (i.e., a textual comment); the annotation body can be a text but also any other digital resource (or its part). The OA model is quite general and does not prescribe how annotations should be presented to users; in particular it is not possible to represent ‘explosive’ annotations stating that a video at a certain time instant should be replaced by another one providing for example details about a topic.

In the above mentioned cases, annotation models are focussed on augmenting scenes with additional descriptors and marginally on establishing relationships among audiovisual as in MyStoryPlayer. On such grounds, these the annotation models do not describe the underling semantic model to be executed during the interpretation of the annotations on the browser/client player. For example, they do not cope with what happens when an audio-video segment is annotated by another video segment and the former has a higher time duration; how to annotate a video in a given time instant to play 10 min of another video by stopping the annotated former video when the latter is played; what happens when the context is changed (the user prefers to follow the story of the annotation and not the story into the annotated video). The answer to these questions can be provided only by combining the annotation formal model with an executable semantics. Moreover, these activities should be accessible online, via a sort of streaming annotations, as in the plan of the OpenAnnotation [41].

2.3 Authoring audiovisual annotations for streaming and web applications

In the literature, several tools for media authoring have been proposed, see for a detailed review [16]. In general, the authoring tools for media annotation range from professional tools to broadcast annotations with videos for Video on Demand (VOD) distribution, to web TV such as 4oD (Channel 4's download service), BBC i-player, Sky+. Users of these annotated videos are mainly passive and are interested only marginally in interacting and changing the stories they are observing. An example is eSports, which is a collaborative video execution and annotation environment [57]. Annotations can be integrated into the video stream to be distributed to final users via RTSP streaming. Annotations can be simple text, audio and geometrical forms and are coded by using MPEG-7. In this case, the audio visual can be used as annotations on other videos and cannot be used for navigation on a multi-stream view, for example to jump on a different context by selecting one of the presented videos. A video annotation tool for MPEG-7 has been proposed also in [40] mainly to describe the content inside the scene. Other video annotation models and tools, such as the IBM VideoAnnEx [50] allow to comment video with static scene descriptors. SMAT [52] allows to annotate video clips with text on whiteboard collaboratively. Hyper-Hitchcock [49] is an interactive environment for authoring and viewing details of on demand videos. It includes an interactive editor to add details on videos through a process of composition and link creation, a hyper video player, and algorithms for the automatic generation of hyper video summaries related to one or more videos, thus allowing to navigate among video chunks. What's Next [48] is a video editing system helping authors to compose a sequence of scenes which tells a story, by selecting each scene from a corpus of annotated clips. Authors can type a story in free English, and the system finds possibilities for clips that best match high-level elements of the story. It operates in two phases, annotation and composition, working with recommendation functionalities. NM2 (New Media for a New Millennium) [25] aims at developing tools for the media industry enabling the efficient production of interactive media. NM2 productions consist of a pool of media assets to be recombined at runtime based on a logical description of the story and the end user's interaction.

On the other hand, in web applications both annotations and video relationships can be dynamically user generated. Simple web based annotation models and players are available in solutions such as the users' annotations on the images of Flickr or YouTube Video Annotation [56] which provides links to navigate among different

related videos. Similar solutions can be obtained in HTML/HTML5 with java script (see for example, popcorn.js). The jump to the next video interrupts the experience of the users. In the latter case, an annotation may allow to select the next scene, to pass to different videos, to model the paradigm of Hypervideo (see MIT Hypersoap [20], OverlayTV [42], etc.). Many authoring tools allow the creation of annotations for the analysis of the narrative of media such as [34, 54]. Simple paths may be created by starting a video from an overlapped link placed on another video. In some cases, the semantic description may also establish time synchronization of the annotation with the annotated audiovisual media. This occurs in *Lignes de Temps* [33], eSports [57], which allows to synchronize simple annotations to a video. This means that during the video play/execution annotations may pop up, be browsed and discussed. In this line, Virtual Entrepreneurship Lab, in short VEL of [28], has proposed video annotations in MPEG-7 on educational activities: they are substantially links to other videos in a graphic page, and the user may select them. In VEL, a video may offer related videos which the user may jump on (restarting it from zero on the central player), and they may be considered as simple annotations. In those cases, the time lines of the video annotations are neither ordered nor related one another, and any video annotation is a stand-alone video. In [37], a XML format for producing non-linear interactive video (presentations where video segments can be activated by users via links, buttons, and on given conditions) with the aim of providing a tool for navigating within a structure has been presented. Also in this case, the model has been proposed to create pre-produced elaborations that cannot be changed by the users enriching the non-linear video model.

Most of the above mentioned annotation tools are mainly oriented to provide a tool for the creation of annotations meant to prepare the video streaming (see for example the above cited: NM2, AXMEDIS, eSport, What's Next) and unable to provide support for the creation of annotations on the client side, which means accessible and selectable by final users via web browsers. The mesh of possible annotations and relationships among audiovisual can be prepared and offered to the user who can navigate in the related media exploiting aside menu or overlapped choices. Despite this large work on annotation tools, most of them are focussed on annotation formats which prevent from defining the player behaviour in executing the annotation (i.e., annotation semantics), nor the progressive delivering of annotations on demand from a client and a server. Moreover, they do not allow the annotation of an audiovisual with another audiovisual segment with the aim of both creating a browseable set of relationships among audiovisual relationships and annotations, and presenting the temporally overlapped audiovisual as synchronous media. In most cases, they are limited in the number of streams and media that can be played at the same time and the usability in terms of context change is limited, as well.

In many applications, the preventive production of annotated media (non-linearly related video segments) by matching media and providing all possible paths is not viable, since this approach could not limit the interactivity and manipulation capabilities of the user (such as the creation and addition of other relationships). Moreover, the activity of media compounding would be regarded as the exploitation of media adaptation rights—e.g., [8]. This right also implies the possibility of modifying the original media which is a formal right quite difficult to be obtained by content owners. Therefore, a web annotation tool has to be able to work on audiovisuals and preserve their integrity [9]. This means to create annotations, as well as the composition and navigation of media relationships and synchronizations, without changing, nor combining in a stream the original media. Therefore, the solution to

create new experiences based on the composition, aggregation and reuse of accessible audio visual can be restricted by establishing relationships that can be executed at the play time without changing the original media.

Another very important aspect is the mechanism to help in identifying the media to be annotated on the basis of the associated metadata. A large amount of audio visual may directly arrive with some pre-established relationships among one another: sequences, multicamera shooting, collections, playlists, etc. For example, it is common to have audiovisual belonging to the same sequence may have title with a common or similar root. For example, "Enrico iV, act.1", "Enrico iv, 2/2".

3 MyStoryPlayer media relationship model

The MyStoryPlayer/ECLAP scenarios need to be grounded by a suitable and expressive annotation model semantically conformant with the notions adopted in the education and training scenarios, where different audiovisual resources are formally related one another, frequently presenting synchronizations. This means to build a formal annotation model which has to provide a precise semantics and behaviour for the different scenarios the player is following during the media playing and its related execution of annotations, and also in the cases of user's interaction and actions (since in most cases this could imply a change of content, even the destruction of the video stream caching). For these reasons, the MyStoryPlayer solution and formal model have to provide support to: (i) create relationships among audiovisual media with multiple views for play and rendering and specific semantic associated with their play (execution of relationships at play time), (ii) jump to a different context (selecting the next central audiovisual at a time instant) and obtain from the server a new set of streamed annotations from the new context, (iii) jump back in the stack of actions, thus keeping trace of the performed context changes, (iv) save and share the user experiences built on the basis of the performed plays and context changes among the audiovisual. The performed navigations (i.e., play, jump, back, swap to another video) among media relationships, executions and specific user actions to change the context (passing from one video to another) allow the user to produce his/her own experience in the navigation and play among audiovisual annotations, that can be saved and shared with other users. Moreover, the MyStoryPlayer makes the production of media relationships in the context of ECLAP social network easier, thus granting the production of a large mesh of browse-able media relationships.

According to the related work and the above description, the peculiar aspects of MyStoryPlayer and differences with respect to the state of the art have been remarked. Each relationship may have a descriptor associated to provide details about the contextual information: the reasons for the relationships, the description of the scene, etc. The MyStoryPlayer approach is quite different from the annotation tools mentioned in Section 2, where annotations to an audiovisual mainly refer to the scene content. In MyStoryPlayer, descriptors are associated with relationships among media and thus with the logic semantics of the established relationships. For example, this audiovisual segment reports "*left view of the central view*", this segment reports the "*Dario Fo actor doing the same gesture but in a different context*", this segment is reporting a "*different actor interpreting the same scene of the previous one*".

In the literature a wide effort has been done to model temporal relationships [2] and temporal logics [7]. These aspects are out of the scope of this paper, while a

temporal notation is used to formalize the temporal relationships among audiovisual segment annotations.

In the following subsections, the main kinds of media relationships and their related meaning at the execution time are presented. In MyStoryPlayer, an audiovisual content has an associated executable timeline with a given duration. This is true for video and audio, while images can be shown for a given time duration. According to the analysis performed for the above presented scenarios, the typical relationships which can be needed to formalize the presentation of audiovisual in a lesson can be described as:

- **Explosion relationship** which consists in associating at a given time instant of a master audiovisual segment the execution of a second audiovisual segment, interrupting the execution of the former. This kind of relationship can be used when the teacher wants to explain a concept during a story, thus opening a digression.
- **Sequential relationship** which formalizes that one audiovisual has to be executed after another one. At the end of a given medium execution, the sequential one will start automatically, changing the context on the player. This kind of relationship can be used when there are many videos parts of the same video; for example, Act 1 and Act 2, lesson 1 and lesson 2.
- **One2One relationship** which consists in relating an audiovisual segment to annotate another audio visual segment, with the aim of showing the former segments synchronously when the latter is executed, and not the opposite.

These relationships among audiovisuals can be matched to create more complex structures and relationships among audio-visuals which can be navigated and executed, as well.

Let us now formalize the MyStoryPlayer model:

$$MSP = \langle Media, O2O, Exp \rangle$$

where:

- $Media = \{M_1, \ldots, M_N\}$ is the set of audiovisual contents (video, audio and images) that are subject to relationships;
- $d(M)$ where $M \in Media$ represents the duration of media M, for images the duration is considered unlimited; On the other hand, images can be rendered according to a specified duration into the defined relationship;
- $M^{[s,e]}$ where $M \in Media, s \geq 0, e \leq d(M)$ represents the section of media M starting at time s and ending at time e;
- $M^{[t]}$ where $M \in Media, t \geq 0, t \leq d(M)$ represents the media frame which can be seen at time t;
- $O2O = \left\{ \left(M_A^{[s_A,e_A]}, M_B^{[s_B,e_B]} \right) \ldots \right\}$ is the set of One2One relationships, where media section $M_A^{[s_A,e_A]}$ is related with media section $M_B^{[s_B,e_B]}$
- $Exp = \left\{ \left(M_A^{[a]}, M_B^{[s_B,e_B]} \right) \ldots \right\}$ is the set of Explosive relationships, where media M_A is exploded at time instant a with media section $M_B^{[s_B,e_B]}$

Moreover two useful projection functions are:

- $O2O[M_a] = \left\{ \left(M_a^{[x_1,y_1]}, M_{a_1}^{[s_1,e_1]} \right), \left(M_a^{[x_2,y_2]}, M_{a_2}^{[s_2,e_2]} \right) \ldots \left(M_a^{[x_n,y_n]}, M_{a_n}^{[s_n,e_n]} \right) \right\} \subseteq O2O$ is the subset of $O2O$ with the One2One relationships related to media M_a;

- $Exp[M_a, s, e] = \left\{ \left(M_a^{[x_1]}, M_{a_1}^{[s_1,e_1]} \right), \left(M_a^{[x_2]}, M_{a_2}^{[s_2,e_2]} \right) \ldots \left(M_a^{[x_n]}, M_{a_n}^{[s_n,e_n]} \right) \right\} \subseteq Exp, s \leq x_1 < x_2 < \ldots < x_n \leq e$ is the subset of *Exp* with Explosive relationships related to media M_a that are within time instants *s* and *e*;

Other useful definitions are:

Definition Media concatenation operator:

$M_A \oplus M_B$ represents the media obtained by concatenating media M_A with media M_B.

Definition Media translation operator:

$M_A|_x$ represents the media obtained by translating media M_A *x* seconds in the future; for example $V_A^{[5,60]}|_{30}$ presents the section from second 5 to 60 of video V_A after 30 s.

In the following subsections, the media relationships are analysed from the point of view of their semantic meaning. Without losing generality, in the following examples, we are talking about video, similar issues can be stated for audio tracks, and images with a given duration.

3.1 Explosion relationship

Explosion relationships aim at expanding the execution time line of a video (e.g., V_1) with the insertion of an identified segment of a second video, V_2; just returning to the execution of V_1 once the V_2 segment execution is completed (see Fig. 1). This model is equivalent to the action of opening a parenthesis where some aspects can be recalled, and then closing it to restart from the point where the parenthesis had been opened. The Explosion relationship can be used to explain a single time instant with an expanded scenario; to show possible cut scenes, to stress possible variants, to insert comments by the director, to explode a single time instant with a more complex scenario, etc.

The Explosive relationship consists in the association of a video time instant of media/video to a second media segment. At the play of V_1, the relationship is executed by starting the play of the second video segment. According to the above model, the screen rendering of an audiovisual medium is a function $\mathbb{M}[M, Exp]$ that given a medium $M \in Media$ and a set of Explosive relationships *Exp* provides the media to be played on the main screen, considering

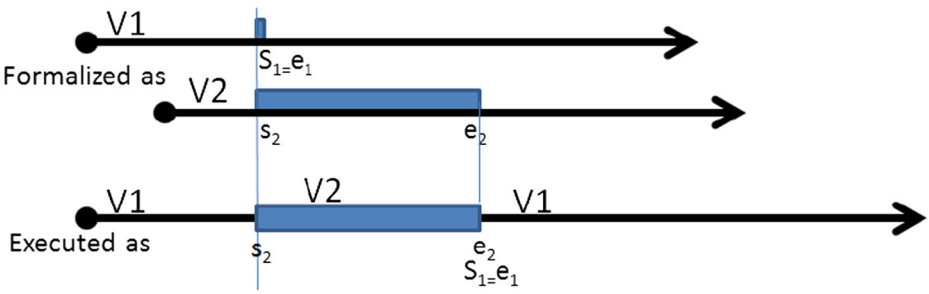


Fig. 1 Explosion relationship: formalized as above and executed as below: at V1.s_1 the player starts reproducing V2 from s_2 until point e_2; then the reproduction switches back to V1, just a time instant after s_1

the explosion relationships that are available in *Exp* regarding media M. Thus, it can be defined using the recursive function $\mathcal{M}$, $\mathbb{M}[M, Exp] = \mathcal{M}[M, Exp, 0, d(M)]$ defined as:

$$\mathcal{M}[M, Exp, s, e] = M^{[s,a_1]} \oplus \mathcal{M}[M_1, Exp, s_1, e_1] \oplus M^{(a_1,a_2]} \oplus \mathcal{M}[M_2, Exp, s_2, e_2] \oplus \ldots \oplus \mathcal{M}[M_n, Exp, s_n, e_n] \oplus M^{(a_n,e]}$$

Where:

- $Exp[M, s, e] = \left\{ \left(M^{[a_1]}, M_1^{[s_1,e_1]} \right), \left(M^{[a_2]}, M_2^{[s_2,e_2]} \right) \ldots \left(M^{[a_n]}, M_n^{[s_n,e_n]} \right) \right\}$ with $s \le a_1 < a_2 < \ldots < a_n \le e$
- considering that: $\mathcal{M}[M,Exp,s,e] = M^{[s,e]}$ if $Exp[M,s,e] = \varnothing$

On the basis of the model, we can see the following example where a set of explosive relationships is are associated with set of media: If $Exp = \{(V_1^{[10]}, V_2^{[15,60]}), (V_1^{[60]}, V_2^{[30,80]}), (V_2^{[40]}, V_3^{[0,30]})\}$, and $d(V_1)=100, d(V_2)=80, d(V_3)=30$, then:

$$\mathbb{M}[V_2, Exp] = V_2^{[0,40]} \oplus V_3^{[0,30]} \oplus V_2^{(40,80]} \quad \text{and}$$
$$\mathbb{M}[V_1, Exp] = V_1^{[0,40]} \oplus V_2^{[15,40]} \oplus V_3^{[0,30]} \oplus V_2^{(40,60]} \oplus V_1^{(10,60]} \oplus V_2^{[30,40]} \oplus V_3^{[0,30]} \oplus V_2^{(40,80]} \oplus V_2^{(60,80]}.$$

3.2 Sequential relationship

Sequential relationships are used to model the sequences of media, for example when different videos are taken from the same event as sequential parts. Thus, there is the need of creating a sequence of audiovisual reproductions, as in the playlists. The associated behaviour of this relationship puts the new video in place of the previous one and loads and shows the set of relationships associated with the new context. The formal definition of this relationship consists in the association with the last time instant of a video to the start of the successive video. The Sequential relationship has not been modelled as a primary operator and set of the MyStoryPlayer model, since it can be derived from the other operators. In this example, in order to sequence video V_1 with video V_2, an explosive relationship is put at the last instant of video V_1 with

$$Exp = \left\{ \left(V_1^{[d(V_1)]}, V_2^{[0,d(V_2)]} \right) \right\},$$

in this case

$$\mathbb{M}[V_1, Exp] = V_1^{[0,d(V_1)]} \oplus V_2^{[0,d(V_2)]} \oplus V_1^{(d(V_1),d(V_1)]} = V_1^{[0,d(V_1)]} \oplus V_2^{[0,d(V_2)]} = V_1 \oplus V_2$$

3.3 One2One relationship

According to the MyStoryPlayer model, it is possible to associate with a specific time instant of an audiovisual content a time segment of other audiovisual content, establishing a One2One relationship. For example, it is possible to relate a video segment (starting from 1′:30″ to 10′:00″ with respect to its beginning) to another video segment (from 2′:15″ to 7′:50″). The possible cases are reported in Fig. 2, where media segments are marked with their start *s* and end *e*, and are aligned to their starting points (from which their synchronous execution has to start), and they are taken from two different media of different length/duration. This model allows the execution of synchronized audiovisual media by starting from a given time instant. For example, to

- compare different performances of the same opera with different or same actors, different years,..;
- compare different interpretations of the same actions or of different actions;
- show/see different points of view or what happens meantime in another place; useful for: presentations, big brother events, sport events, theatrical shows, related historical scenarios, etc.;
- show a video with the scene without visual effects; useful for backstage presentations in training and for infotainment;
- remind of past related events, or link to possible future events; useful for aside advertising;
- provide the comment of the director;
- etc.

The formal definition of the relationships consists of the association of a video segment of V_1 to a video segment V_2. At the play of video V_1, the relationship is executed by starting the play of V_2 from s_2. The V_2 segment starts synchronously playing from s_2 time instant only when its master V_1 is put in execution and reaches s_1 time instant. The determined relationship between the two videos is asymmetrical since *relating* V_1 *with* V_2 is semantically different with respect to relate V_2 *with* V_1. During the execution, the temporal relationships (time segments), as well as the video screens, are proposed to the user, so as to allow him/her to change the context.

A specific behaviour of the player occurs in the different cases reported in Fig. 2:

- **Case a)** during playing the screen of V_2 is frozen at time instant e_2, then the screen of V_2 is removed to give space to other relationships.
- **Case b)** during playing at e_1 time instant the synchronous execution of V_2 stops. If the user wants to watch the entire V_2 segment, he has to switch to it by clicking on the segment or related audiovisual; thus loading a new context based on V_2 and allowing the reproduction of segment s_2,e_2. Then the screen of V_2 is removed to give space to other relationships.
- **Case c)** this case corresponds to synchronized play of the two video segments.

The definition of the screen rendering for this One2One relationship is defined as function $\mathbb{S}[M, O2O]$ which, given a medium $M \in Media$ and a set of One2One relationships $O2O$,

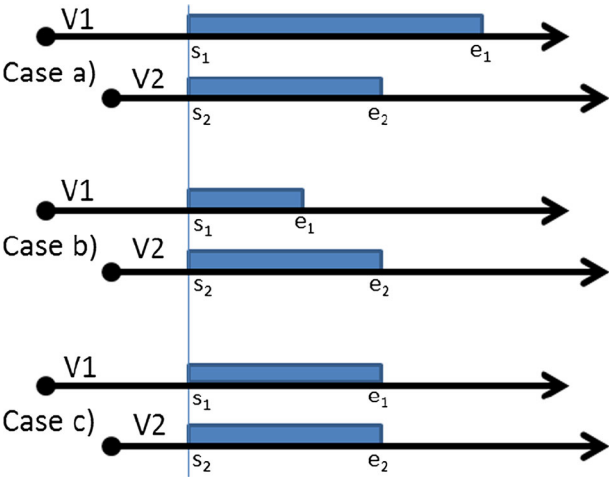


Fig. 2 One2One relationships between two audiovisual, the main cases

provides the set of media to be played on the side screens when medium M is played on the main screen:

$$\mathbb{S}[M, O2O] = \left\{ M_1^{[s_1,\min(e_1,s_1+y_1-x_1)]}\Big|_{x_1} \ldots M_n^{[s_n,\min(e_n,s_n+y_n-x_n)]}\Big|_{x_n} \right\}$$

Where:

$$O2O[M] = \left\{ \left(M^{[x_1,y_1]}, M_1^{[s_1,e_1]} \right), \left(M^{[x_2,y_2]}, M_2^{[s_2,e_2]} \right) \ldots \left(M^{[x_n,y_n]}, M_n^{[s_n,e_n]} \right) \right\}$$

It should be noteworthy that the video is translated to the starting point of the annotation x_i and that the end of the annotation media is updated to $\min(e_i, s_i+y_i-x_i)$ in order to view the target media M_i for the duration of the annotation (y_i-x_i) or until the end of the target video section if less.

According to the above cases, an example including both Case (a) and (b) with:

$$O2O = \left\{ \left(V_1^{[10,15]}, V_2^{[20,35]} \right), \left(V_1^{[40,70]}, V_3^{[80,90]} \right) \right\}$$

We have that the side screen video for V_1 are

$$\mathbb{S}[V_1, O2O] = \left\{ V_2^{[20,25]}\Big|_{10}, V_3^{[80,90]}\Big|_{40} \right\}$$

Then two videos, V_1 and V_2, are synchronously executed when the V_1 is played (Case (c)):

$$O2O = \left\{ \left(V_1^{[0,d(V_1)]}, V_2^{[0,d(V_2)]} \right) \right\}, \text{with} \, d(V_1) = d(V_2)$$

We have

$$\mathbb{S}[V_1, O2O] = \left\{ V_2^{[0,\min(d(V_1),d(V_2))]}\Big|_0 \right\} = \left\{ V_2^{[0,d(V_2)]}\Big|_0 \right\} = \{V_2\}$$

However, if for a medium both explosive and One2One relationships exist, then the medium for the side screens depends on the video section currently played on the main screen which can be a section of a medium, e.g. $M^{[s,e]}$, for this reason the Side Screen presentation function needs to be extended as follows.

$\mathbb{S}'\left[M^{[s,e]}, O2O\right]$ is a function which, given a medium section [s,e] of $M \in Media$ and a set of One2One relationships $O2O$, provides the set of media to be played on the side screens when playing the medium section.

Considering the one2one annotations that are available on medium M

$$O2O[M] = \left\{ \left(M^{[x_1,y_1]}, M_1^{[s_1,e_1]} \right), \left(M^{[x_2,y_2]}, M_2^{[s_2,e_2]} \right) \ldots \left(M^{[x_n,y_n]}, M_n^{[s_n,e_n]} \right) \right\},$$

only the annotations that are active in time interval [s,e] should be considered and should be limited to the section being annotated to cover the [s,e] interval, thus considering that:

$$[x_i, y_i] \bigcap [s, e] = [\max(x_i, s), \min(y_i, e)] = \left[\overline{x}_i, \overline{y}_i \right]$$

we have

$$O2O\left[M^{[s,e]}\right] = \left\{ \ldots \left(M^{\left[\overline{x}_i, \overline{y}_i\right]}, M_i^{\left[s_i+\overline{x}_i-x_i, e_i\right]} \right), \ldots \right\} \text{considering only terms having} \, \overline{x}_i \le \overline{y}_i$$

where the starting point of the target media is updated to $s_i + \overline{x} - x_i$ in order to consider the case when s is in the middle of interval $[x_i, y_i]$ and thus the part of target media for the duration from x_i to s has to be skipped.

Finally we have that:

$$\mathbb{S}'\left[M^{[s,n]}, O2O\right] = \left\{ \ldots M_i^{[s_i+\overline{x}_i-x_i, \min(e_i, s_i+\overline{y}i-x_i))]}\Big|_{\overline{x}_i-s} \ldots \right\} \text{ considering only terms having } \overline{x}_i \leq \overline{y}_i$$

Should be noteworthy that the media translation points are provided with respect to time instant s.

For example if

$$O2O = \left\{ \left(V_1^{[10,20]}, V_2^{[20,60]}\right), \left(V_1^{[40,70]}, V_3^{[10,90]}\right), \left(V_1^{[60,80]}, V_4^{[10,50]}\right), \left(V_1^{[90,110]}, V_5^{[0,40]}\right) \right\}$$

$$\mathbb{S}'\left[V_1^{[50,100]}, O2O\right] = \left\{ V_3^{[20,40]}\Big|_0, V_4^{[10,30]}\Big|_{10}, V_5^{[0,10]}\Big|_{40} \right\}$$

3.4 Reciprocal synchronization relationships

The above presented models can be combined to create more complex relationships of synchronization in the presence of multicamera. In MyStoryPlayer, the Reciprocal Synchronization relationship implies a couple of relationships—e.g., *from* V_1 *to* V_2 *and viceversa*. This relationship can be useful in the event of a video related to same issue and with the arising need to synchronize items (different aspects, different point of views, etc.). There are many contexts where this can be applicable like for example in the didactical framework of theatre lessons. It is useful to have different points of view of the scene and have them played in a synchronized way; but also when dealing with performances and sport related events.

The described relationship of synchronization can be defined among several audio-visual (e.g., 8 camera views of the same events). In the case of N videos a total of N^2-N relationships are defined. This allows to see synchronized camera regardless of the starting point. This allows the access to the full set of videos relationships by playing any of the involved videos. This kind of formalization can be useful to synchronize multiple views, regardless of the video which has been selected to play. Similarly to the One2One type, the lengths of synchronized media segments can be chosen independently. What is different with respect to One2One is the nature of the relationships which is mutual (in both sense) and can involve N media.

4 MyStoryPlayer tool, user interface features

The exploitation of the above described relationships consists in the media playing and thus synchronous execution of different videos, audios, and images. MyStoryPlayer supports the execution of complex relationships among audiovisual of different kinds: One2One, Explosion, Synchronization and Sequential.

On ECLAP portal there are many groups of videos related to the same event, divided into sequential parts and taken in multicamera, related one another to create a playable structure where the user can navigate view/play in an interactive and synchronized way through MyStoryPlayer facility. Figure 3 shows a simple example of synchronous and sequential relationships among videos. This example refers to a

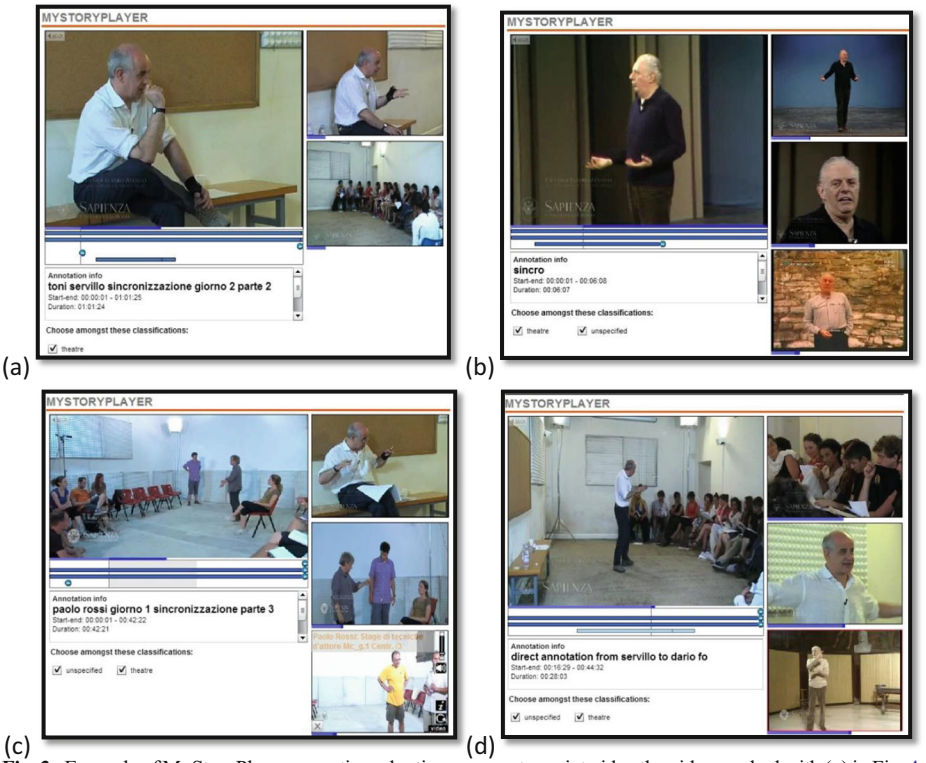


Fig. 3 Example of MyStoryPlayer execution adopting as a master point video the video marked with (*a*) in Fig. 4. The screenshots depict the user point of view in different situations met during the navigation among relationships and highlighted into Fig. 4. **a** Two reciprocal synchronization relationships, one Exp and a One2One relationship; **b** the explosive annotation jumped on the master position, becomes active and the scenario changed, going to Dario Fo synchronization of Miracolo di Gesu Bambino play; Once the explosive annotation is terminated the context returned in the context of (**a**); **c** the user clicked on the One2One relationships in (**a**) going to the Paolo Rossi's Theatrical Lab.; The *grey zones* overlapped on the timelines represent the length of relationships that are played for that medium; **d** the user came back in (**a**) situation from (**c**) and clicked on a video on left, thus changing the context by swap to new scenario (**d**) with a direct annotation to Dario Fo

segment of the relationships established to model the theatrical laboratory by Toni Servillo, an activity which took place at CTA Rome. This laboratory lasted several days and it has been recorded from three cameras (right side, frontal and left side), the video recording of each day is divided into many parts. In Fig. 3, an intuitive representation is provided, the column of videos represents the videos taken at the same time (from left, centre and right camera, sometimes close-up shooting) while the raw of videos represent the successive recordings, which correspond to time (part, lessons, acts, days), etc. The established relationships create a playable structure ordered in time by sequence and synchronized by point of view. The result is a multi-views vision for each part of the sequence, as depicted in Fig. 4.

In Fig. 4, letters from (a) to (c) are indicating some entry points (examples of links that can be provided by the teacher to start playing the lesson). They are a way to access into the non-linear structure of relationships that may be executed by the MyStoryPlayer. Users may start playing the above structure of media by entering into

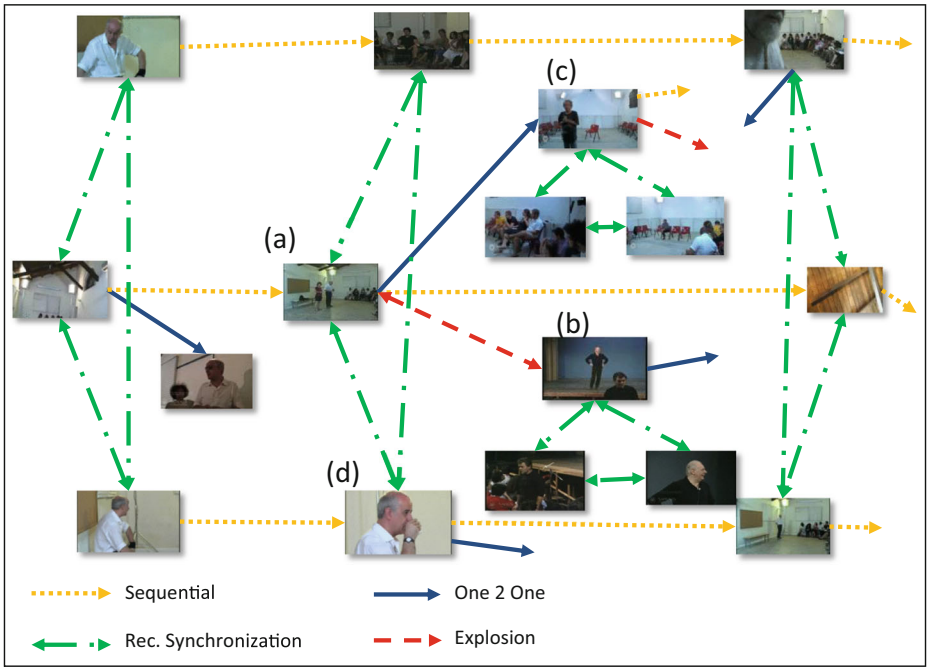


Fig. 4 Example of relationships among several media executed by the MyStoryPlayer (*a part*). The *letters* identify the scenarios corresponding to the snapshots commented in Fig. 3

it via anyone of the above media, as a click from a query result. For example, via the video labelled (c); this is possible by accessing the MyStoryPlayer and using the small icon on that video 3 http://www.eclap.eu/drupal/?q=en-US/msp&axoid=urn:axmedis:00000:obj:1fd0220e-36c6-4df6-8b04-e38903d0759f&axMd=1&axHd=0 within the ECLAP portal, in any list of query results where that video appears. Starting from the (c) media, the MyStoryPlayer is going to put as main video central video/part 1 (first day), playing the other synchronized videos (left and right) aside the main one, as depicted in Fig. 3. According to the sequential relationships, at the end of the main video the execution begins to load the synchronized videos of the second part, and so on.

According to Fig. 3, the user can see on the left side the master medium. The related audiovisuals according to One2One relationships are dynamically listed on the right side: for example the left and right camera, the next explosive media to be played, etc. According to the above discussed relationship model and semantics, the media are streamed synchronously with the master media. The set of relationships among the involved videos is reported in Fig. 4, the corresponding snapshot is reported in Fig. 3. For example, in Fig. 3a, under the master video the time bar reports the streaming progress and depicts the relationships with the other media: Explosive and One2One relationships (respectively depicted as circle in the timeline and as a segment inside a timeline) and as a Sequential relationship reported at the end of the time line as a small circle. Synchronizations are additional bars on the main timeline.

On the MyStoryPlayer the user interface (which is partially activated by moving the mouse over the master), the user may:

- click on the master time line to jump forward and backward on the play time;
- click on one of the right side media to swap it on master position, thus changing the media of the main context, which also implies the visualization and activation of the relationships associated to that medium;
- click on the back button on the upper left corner to swap back;
- start, pause and stop the execution of the master medium and thus of the whole rendering;
- start and stop the experience recording;
- activate and/or regulate the audio coming from different media sources (audio and video) which are synchronously executed;
- select labels (below the master position and timeline) and thus highlight the media which he/she is interested in, among those in execution on the right side;
- move the mouse over the different media to see their titles and other pieces of information;
- move the mouse over the timelines depicted for the different media to see the descriptions of the related media and identify them on the right side list.

The example of Figs. 3 and 4 focuses only on a possible path which the user can follow in this relationship structure. As we can see from Fig. 4, there are many possible paths the user can follow in his experience on MyStoryPlayer. This approach is quite similar to browsing and navigation experience on web pages' hyper-links. The more the relationships grow, the more the user is provided with paths to navigate on. *Moreover, the performed navigation can be saved and shared with other users as depicted in next subsection.*

4.1 Exploiting relationships to navigate on audiovisual media by using MyStoryPlayer

In this section, an example about the use of MyStoryPlayer to perform and save an experience in navigating among relationships as shown in Fig. 4 is provided. When the user starts executing the MyStoryPlayer from the media indicated by (a) in Fig. 4 (e.g., (a) snapshot of Fig. 3), the MyStoryPlayer Client loads the scenario with the main video synchronized with other two videos, with an explosive annotation, a direct annotation in a specific segment of the timeline, and a related video in sequence. Therefore, the user has the opportunity to drop the execution of the master medium by jumping on a different medium/context and to follow one of the proposed relationships, and thus the new media and related relationships are executed synchronously, as well. The user may switch from one video/medium to another and return back in the stack of events, by using the BACK button. From that point, in Fig. 5, a possible user navigation is reported as described in the following points, executing a set of segments:

1. from (a) context, the MyStoryPlayer reproduces synchronously the main medium with the other two. The user started the recording at a time instant in the context of the first medium of *Toni Servillo*. The user could choose to swap to another related medium. In this case, the user keeps on viewing the medium with original title "*Servillo: Stage di Tecniche d'attore Mc g.2 centr g.2 dx/2*";

2. at a specific time, according to the Explosive relationship the context is shifted to the related media (thus passing to (b)). In (b), three synchronized media (a master plus other two) (see (b) on Fig. 3, *Il primo Miracolo di Gesù bambino*) are provided along the duration of the relationship. The duration is identified with the dark zone 2 in Fig. 5, where a One2One relationship is included. Once zone 2 has been played, the MyStoryPlayer returns back automatically to context (a). The MyStoryPlayer and the Explosive relationship bring back the user to the previous scenario (a) to execute segment 3;
3. in segment 3 of Fig. 5, the scenario has 2 Synchronizations and a One2One relationship. During the viewing of the media and in the context (a), the master has the original title as "*Servillo: Stage di Tecniche d'attore Mc g.2 centr g.2 dx/2*". The user decides to view the video related with One2One relationship by clicking on video (c) on the right side. This happened at the time instant marked with a yellow star on segment 3. The user changed the context by swapping, thus passing to context (c) and placing that video to the master position;
4. the execution is shifted to play segment 4 in (c). The user could view the video for the entire duration, or can go back to the previous scenario. On the other hand, after a while, the user goes back, by click on the Back buttons. Therefore the player brings the user back to the end of the relationship in the former video, to execute segment 5;
5. segment 5 is executed for a while, then the user decides to perform a jump forward on the timeline of video (a), thus moving to segment 6;
6. During the execution of segment 6, the user performs a swap by selecting video (c) on the right side of the user interface. The new context has some synchronizations and annotations;
7. On segment 7, the user decides to play for a while. Then, the recording experience is stopped by the user and the experience can be saved.

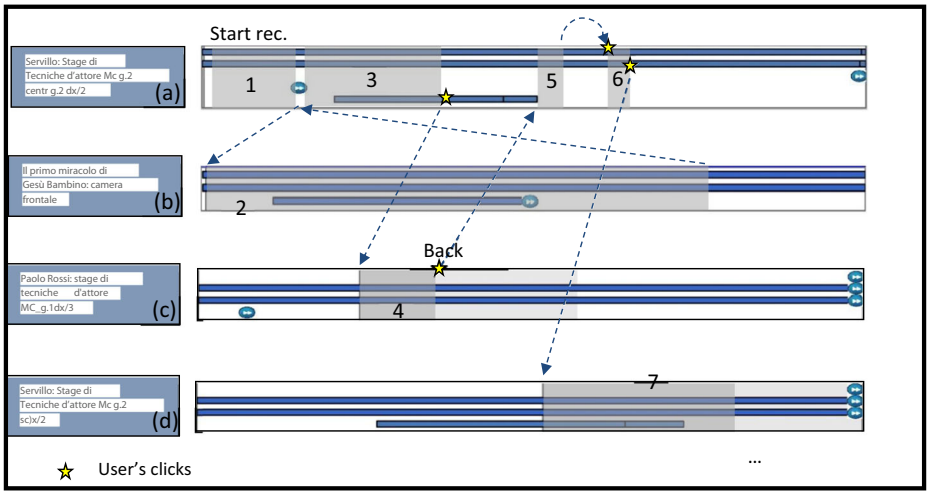


Fig. 5 Example of user navigation among several media relationships

The saved experience can be formalized as follows, using RDF-XML and recording only the actions done by the user:

```
<msp:Experience rdf:about="http://www.eclap.eu/msp/experience/Exp1">
        <dc:title> Test for Paper</dc:title>
        <dc:description>Experience for Paper</dc:description>
        <dc:date>2013-10-28</dc:date>
        <msp:hasFirstStep rdf:resource=" http://www.eclap.eu/msp/experience/Step1"/>
</msp:Experience>
<msp:Begin rdf:about="http://.../Step1">
        <msp:isFirstStepOf rdf:resource="http://.../Exp1" />
        <msp:hasNextStep rdf:resource=" http://.../Step2"/>
        <msp:mediaUri rdf:resource="urn:axmedis:00000:obj:b44bad4b-817c-43ac-8553-8b83d5764a0b"/>
        <msp:clickTime>00:04:56</msp:clickTime>
</msp:Begin>
<msp:Swap rdf:about="http://.../Step2">
        <msp:isStepOf rdf:resource=" http://.../Exp1" />
        <msp:hasNextStep rdf:resource =" http://.../Step3"/>
        <msp:swapAnnotation rdf:resource="http://...#Relation_1333215764296_1493" />
        <msp:timeSwap>00:18:57</msp:timeSwap>
</msp:Swap>
<msp:Back rdf:about="http://.../Step3">
        <msp:isStepOf rdf:resource="http://.../Exp1" />
        <msp:hasNextStep rdf:resource ="http://.../Step4"/>
        <msp:backTime>00:20:12</msp:backTime>
        <msp:backTo rdf:resource="urn:axmedis:00000:obj:b44bad4b-817c-43ac-8553-8b83d5764a0b"/>
        <msp:backToTime>00:31:01</msp:backToTime>
</msp:Back>
<msp:Seek rdf:about="http://.../Step4">
        <msp:isStepOf rdf:resource="http://.../Exp1" />
        <msp:hasNextStep rdf:resource =" http://.../Step5"/>
        <msp:clickTime>00:34:14</msp:clickTime>
        <msp:seekToTime>00:36:27</msp:seekToTime>
</msp:Seek>
<msp:Swap rdf:about="http://.../Step5">
        <msp:isStepOf rdf:resource="http://.../Exp1" />
        <msp:hasNextStep rdf:resource =" http://.../Step6"/>
        <msp:swapAnnotation rdf:resource="http://...#Relation_453748659348_2367" />
        <msp:timeSwap>00:37:42</msp:timeSwap>
</msp:Swap >
<msp:End rdf:about="http://.../Step6">
        <msp:isStepOf rdf:resource="http://.../Exp1" />
        <msp:clickTime>00:49:06</msp:clickTime>
</msp:End>
```

5 Semantic model of MyStoryPlayer

As described in the previous sections, some scenarios and solutions are focused on associating formal descriptors to audiovisual content, so as to describe the included scene (the main aim is to allow the indexing and retrieval of the scenes). In MyStoryPlayer, the focus is on the relationships among media content. The semantics of relationships among media describes some temporal associations among them for the purpose of their playing/execution. This approach can be very useful for didactical

purposes, to compare, to explain, to annotate with another audiovisuals, etc. The navigation among the relationships may describe a story: the aim of director, the linear time, the activities of a given actor, the movements of an object, etc. Therefore, the MyStoryPlayer may be used to create a huge amount of new personal experiences with the same content pool, accessing it from a given point and from that point navigating to an indefinite network of relationships. In order to allow the replication of the experience, the MyStoryPlayer permits both the recording and the sharing of such experiences.

The main idea of MyStoryPlayer is to provide final users with a tool for creating and navigating in a mesh of relationships, while granting to the user full access to the media, which may refer to a set of relationships with their related segments. These relationships may be: (i) labelled to allow grouping and selecting/deselecting them, (ii) commented to index and search them. Moreover, the single medium (audio, video and image) used in MyStoryPlayer has its own classification in the ECLAP archive based on ECLAP model which includes multilingual: Dublin Core, performing arts metadata, aggregation information, tags, taxonomical classification, and technical metadata [11]. To this end, Fig. 6 reports the semantic model of MyStoryPlayer tool and RDF database. RDF navigation and queries allow the extraction of the relevant segment of knowledge to be shown and executed by the MyStoryPlayer tool. The relationship has a description which can be textual or may include additional ontology to describe facts by using more accurate, complete, and specific information for domain and contexts. In ECLAP, that part is implemented as free text which can be searched.

Moreover, the flexibility of the RDF model can lead to the creation of customizable part of description in the form of semantic model, depending on the environment and application use. The MyStoryPlayer model has not implemented this kind of part in the semantic model yet,

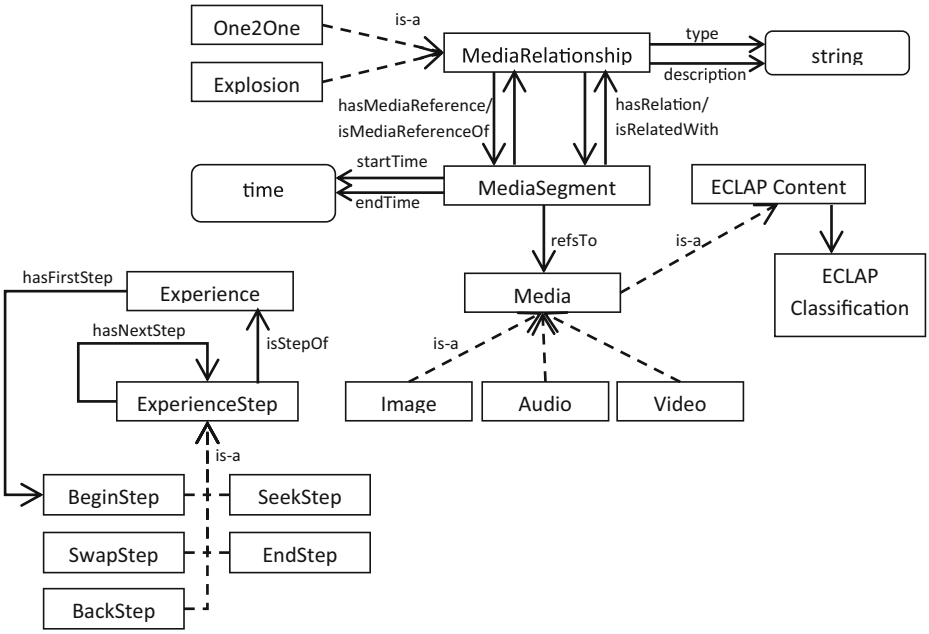


Fig. 6 Semantic model of MyStoryPlayer

since many other tools can be used to this end. MyStoryPlayer is open to accept additional contextual descriptions of scene.

Hereafter the description of the classes using the Manchester OWL syntax is reported:

MediaRelationship = (*isRelatedWith* exactly 1 *MediaSegment*) and
(*hasMediaReference* only *MediaSegment*) and
(label max 1 string) and
(description max 1 string) and
(*createdBy* exactly 1 User) and
(createdAt exactly 1 dateTime)
MediaRelationship disjointUnionOf *One2One*, *Explosion*
MediaSegment = (*refsTo* exactly 1 *Media*) and
(startTime max 1 time) and
(endTime max 1 time)
Audio, Video, Image SubClassOf *Media*
Media SubClassOf (*ECLAPContent* and (duration exactly 1 *time*))
Experience = (*hasFirstStep* exactly 1 *ExperienceStep*) and
(title exactly 1 *string*) and
(description exactly 1 *string*) and
(date exactly 1 *date*)
ExperienceStep =(*hasNextStep* max 1 *ExperienceStep*) and (*isStepOf* exactly 1 Experience)
ExperienceStep disjointUnionOf *BeginStep,SwapStep, BackStep, SeekStep, EndStep*
BeginStep = *ExperienceStep* and
(*isFirstStepOf* exactly 1 *Experience*) and
(*mediaUri* exactly 1 Media) and
(clickTime exactly 1 time)
SwapStep = *ExperienceStep* and
(*swapRelation* exactly 1 *One2One*) and
(swapTime exactly 1 time)
BackStep = *ExperienceStep* and
(*backTo* exactly 1 *Media*) and
(backTime exactly 1 time) and
(backToTime exactly 1 time)
SeekStep = *ExperienceStep* and
(clickTime exactly 1 time) and
(seekToTime exactly 1 time)
EndStep = *ExperienceStep* and
(clickTime exactly 1 time)

Each audiovisual medium may have multiple relationships, with each of them having their own reference to other media segments, labels and description. Media relationships are divided into two specializations: One2One, Explosion. These relationships are semantically different and they are interpreted differently by the MyStoryPlayer, thus providing users with more flexibility in both creation and navigation phases. The model includes also the part related to the user experience representation which can be saved and retrieved.

The relationships are made available also as linked data using the OpenAnnotation model as reported in [4]. The media relationships are mapped to OpenAnnotation, in particular the 'isRelatedWith' property is mapped to the OA *hasTarget* property representing the medium (or its part) the annotation/relationship is associated with. On the other hand, the 'hasMediaReference' property is mapped to the *hasBody* property as it points out to the associated media describing the main media being annotated. Moreover to indicate the kind of relationship (One2One or Explosive), an additional rdf:type indication is provided. Below there is an example of a relationship/

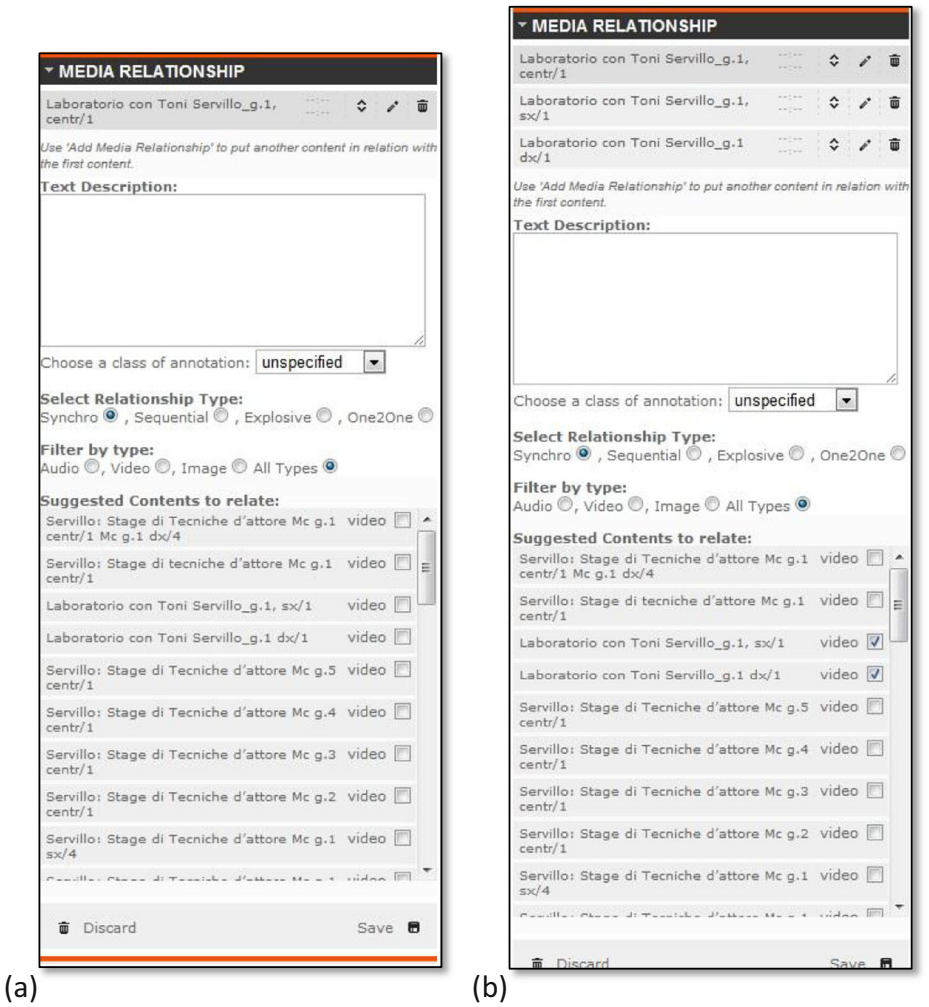


Fig. 7 Creation of assisted relationships among media. The user may select the kind of relationship to be exploited: One2One, Synchronous, Sequential, or Explosive. For example, when selecting 'synchronize', a list of suggested media to be related is proposed (by clicking on the check box related to items, the medium is added to the top list). Then saving the selection implies creating all the mutual relationships. The list of suggested media is produced by a similarity algorithm based on metadata similarity. Different algorithms are used in different cases and the user may filter the results for different media: only video, audio, images, etc

annotation available through Linked Open Data linking a medium from time 29 to 227 with a One2One relationship with another media from time 67 to 119.

```
<rdf:RDF …>
  <oa:Annotation rdf:about="http://www.eclap.eu/resource/annotation/SideAnnotation_136...">
    <rdf:type rdf:resource="http://www.eclap.eu/schema/eclap/One2One"/>
    <oa:hasTarget>
      <oa:SpecificResource>
        <oa:hasSource rdf:resource="http://www.eclap.eu/resource/object/urn:axmedis:00...1"/>
        <oa:hasSelector>
          <oa:FragmentSelector>
            <rdf:value>t=npt:29,227</rdf:value>
            <dcterms:conformsTo rdf:resource="http://www.w3.org/TR/media-frags/"/>
          </oa:FragmentSelector>
        </oa:hasSelector>
      </oa:SpecificResource>
    </oa:hasTarget>
    <oa:hasBody>
      <oa:SpecificResource>
        <oa:hasSource rdf:resource="http://www.eclap.eu/resource/object/urn:axmedis:00...2"/>
        <oa:hasSelector>
          <oa:FragmentSelector>
            <rdf:value>t=npt:67,119</rdf:value>
            <dcterms:conformsTo rdf:resource="http://www.w3.org/TR/media-frags/"/>
          </oa:FragmentSelector>
        </oa:hasSelector>
      </oa:SpecificResource>
    </oa:hasBody>
    <oa:hasBody>
      <cnt:ContentAsText>
        <cnt:chars>this is really interesting</cnt:chars>
        <dc:format>text/plain</dc:format>
      </cnt:ContentAsText>
    </oa:hasBody>
    <dc:type>acting style</dc:type>
    <oa:annotatedBy rdf:resource="http://www.eclap.eu/resource/user/229"/>
    <oa:annotatedAt>2013-02-13T09:32:09</oa:annotatedAt>
  </oa:Annotation>

</rdf:RDF>
```

The original description of the relationship using MyStoryPlayer model is:

```
<rdf:RDF …>
  <msp:One2One rdf:about="http://...#Relation">
    <msp:type>acting style</msp:type>
    <msp:description>this is really interesting</msp:description>
    <msp:createdBy rdf:resource="http://www.eclap.eu/resource/user/229" />
    <ms:createdAt>2013-02-13T09:32:09</msp:createdAt>
    <msp:isRelatedWith>
      <msp:MediaSegment>
        <msp:refsTo rdf:resource="urn:axmedis:00…1" />
        <msp:startTime>00:00:29</msp:startTime>
        <msp:endTime>00:03:47</msp:endTime>
      </msp:MediaSegment>
    </msp:isRelatedWith>
    <msp:hasMediaReference>
      <msp:MediaSegment>
        <msp:refsTo rdf:resource="urn:axmedis:00…2" />
        <msp:startTime>00:01:07</msp:startTime>
        <msp:endTime>00:01:59</msp:endTime>
      </msp:MediaSegment>
    </msp:hasMediaReference>
  </msp:One2One>
</rdf:RDF>
```

6 Production of media relationships

As explained in the previous section, MyStoryPlayer allows the playing of established relationships among media. These relationships can be produced manually, automatically and semi-automatically. The automated production is performed during the content ingestion process of ECLAP and includes the possibility of defining aggregation relationships among media and content according to the tags provided in the metadata by content providers [9].

This section introduces the manual and semiautomatic production processes to create relationships. According to the relationship model as presented in Section 4, a tool has been developed and integrated into ECLAP for registered users. The first step to access the Add Relationship tool and define a new relationship is to select the "Add Relationship" from an audiovisual content in any ECLAP list of content, including content in results of queries, content featured, last posted, top rated, etc. The addition of relationships is part of the typical work teachers would like to perform with their presentation and content organization, thus fully exploiting the MyStoryPlayer model.

Once a user decides to add a relationship, the media relationship tool (depicted in Fig. 7) is proposed, always referring to the selected content (in the same manner other content items can be added, while the first selected content is considered as the master). The interface provides information to the user, and allows to decide which kind of relationship has to be chosen: One2One, Synchronous, Sequential, or Explosive. The ECLAP portal has a large amount of content, users may use the search facilities of ECLAP to identify the content to be related and aggregated according to those models.

Moreover, for the production of Synchronous and Sequential relationships, due to the large amount of content, the Add Relationship tool on ECLAP portal provides suggestions according to the: (i) metadata of the selected master content, (ii) type of relationship model chosen by the user, (iii) type of requested media (video, audio images). The performed estimation of similarity is based on metadata, since in most cases content which can be related via Synchronous and/or Sequential relationships shares similar title, description, text body and/or subject. The solution has been developed by using indexing facilities and exploiting ECLAP classification model as depicted in [11]. See for instance the example reported in Fig. 7a where a number of content has been suggested.

In any case, the user can look for specific content by using the general search facilities of the ECLAP portal. Once the content has been identified, it can be added/selected to have it moving on the top of the Add Relationship tool as depicted in Fig. 7b. When clicking on the checkbox aside the title, the content is added to the list of media to be related. Moreover, it is possible to add to each relationship a textual description and a label about the type of annotation being just created. This is useful when annotations are viewed on MyStoryPlayer in order to distinguish and clarify the type and the meaning of possible relationships related to each audiovisual.

The Add Relationship tool provides a different behaviour according to the type of relationship selected by the user, as described in the following. For example, as to One2One relationship, the involved media are two. The ECLAP audiovisual player allows the definition of the start and end positions on their timeline for audio and video, while for images the duration is requested (see Fig. 8). In the event of Explosion relationships, the user has to identify only a single point on the master audiovisual, and a segment in the second (the audio visual player allows the definition of those points with simple clicks on the timeline). As to Reciprocal Synchronization relationships, the number of media involved can be higher, with N media N^2-N relationships are produced automatically by the tool. For sequential relationships,

Fig. 8 Editing time segment along the temporal line of a video by positioning the cursor in the selected time instant and clicking on the corresponding marker points for start [- and for end -]

the connection of the last time instant of the previous medium with the next one is automatically produced.

The collected relationships for any media can be provided by different users (labelled in different manner, for example) and a set of relationships keeping together a set of media can be fully unconnected with respect to another set of relationships, thus creating several different groups of media, sadly unconnected and strongly related one another, on the basis of their formalized rationales: synchronisation, annotations, sequential, etc.

7 MyStoryPlayer architecture

According to the above-presented scenarios, the user starts accessing the MyStoryPlayer on ECLAP portal for example: (i) by clicking on a small icon highlighting the presence of annotations/relationships associated to a given medium, (ii) by deciding to add a new relationship, (iii) by performing a query on the annotations. Once a relationship is selected, the chosen master medium is put in execution with its related digital media essence, thus the MyStoryPlayer automatically displays a set of relationships and timeline. Relationships are executed aside the main essence according to the timeline and to their inner semantics (One2One, Sequential, Explosive, Synchronization).

In Fig. 9, the general architecture of MyStoryPlayer is provided in relationship to the ECLAP service portal [http://www.eclap.eu]. The MyStoryPlayer Server can be integrated with complex social networks such as ECLAP Social Service Portal (http://www.eclap.eu). In this case, ECLAP users adopt an integrated module (Media Relationship Tools, already described in Section 6) to establish relationships and realize annotations among media, while exploiting the content indexing to produce suggestions for any relationships creation. The established relationships are stored into an RDF database according to the ontology presented in Section 5, thus creating the ontological knowledge database managed by SESAME [15, 46]. The relationships refer to the media essences ingested and managed by the social network via content delivering service (based on xmoov streaming server to enable the seeking forward on server side).

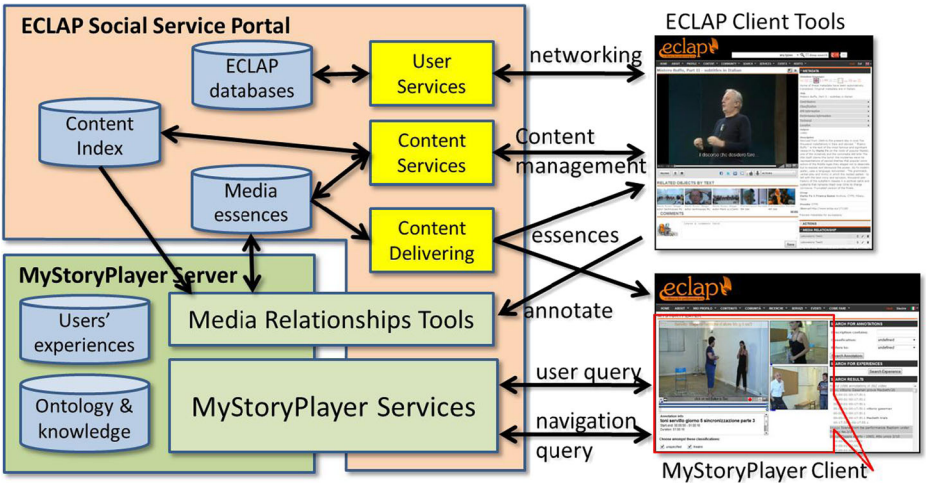


Fig. 9 General architecture of MyStoryPlayer and its integration with ECLAP service portal

When the user opens content into the MyStoryPlayer page, he may perform a semantic query to the semantic database. As a result of the semantic query in SPARQL [51], a list of entry point media and their related range of relationships are provided. An Entry Point scenario may be a video, which is a starting point and from that a set of relations and annotations connected to the video has to be placed in execution.

Moreover, the MyStoryPlayer client tool enforced into the web page is activated on the basis of a medium, and thus it creates a direct connection to the MyStoryPlayer Services to download progressively the set of RDF relationships associated to the master medium every time the master medium changes. For example, whenever the MyStoryPlayer client is forced to change context depending on user interactions. In this case, the change of context also implies to restart progressive download of media streams.

7.1 MyStoryPlayer client

The MyStoryPlayer Client tools have been designed and developed to execute the model presented in Section 3. It has been developed in Adobe ActionScript, so that the player in Flash is automatically provided and loaded during the web page loading as a flash player tool. It could be realised in: SMIL providing an ActiveX including the Ambulant interpreter as in AXMEDIS [8], extending the JavaScript solution as in [23], and probably in HTML5 and JavaScript but with some complexity in managing the synchronisation as discussed below. In any case the problems to be solved remain unchanged:

- the progressive download and interpretation of the media relationships in RDF. As to progressive download, what is meant is the limited download of the information and relationships needed to manage the activated context, and perform the successive download every time the context is changed. It is possible to load in advance the next possible RDF descriptors without waiting for the change of context. This can make the change of context quicker and perform a partial semantic reasoning on the client side;
- the precise management of media synchronizations as described in the assessment and validation section of this paper. The problems of media synchronization occur whenever

the MyStoryPlayer Client is executed, at each change of context: jump, swap, back, etc., and also when One2One relationships become active in the timeline. On the other hand, they are more relevant in the event of low bandwidth. The knowledge of the relationships structure may be exploited to implement preload strategies for the involved media;

- the management of user driven context changes such as jump, swap, back, etc.; the new context has to be loaded and interpreted to be put in execution on the tool. In this case, the change of context also implies to restart progressive download of media streams.
- the management of user interactions and controls on the user interface: volume on multiple media, play/pause/stop of the master which has to control also the other media, independent on/off of audio for each medium involved in the synchronous rendering (thus allowing the listening to the talk of a teacher when slides are shown, hearing and watching director interviews and comments together with the scene, watching multiple scene views of the same event, etc.), and other minor controls;
- the recording of user actions according to Section 4.1, thus managing the modelling, save and load of user experiences with the MyStoryPlayer Services;
- the possibility of rendering master media from different resolutions according to the available bandwidth, also taking into account the number of simultaneous videos to be played and their bit-rate;
- the rendering of textual comments and other information, the management of labels, etc. Textual descriptors to single annotations and resources can be used to highlight specific aspects and to start discussions.

When it comes to implementing some of these major features with the above mentioned technical tools (SMIL in JavaScript, SMIL/Ambulant, HTML5 and JavaScript, and Flash/ActionScript), some limitations have been detected in terms of interactivity and fast response. For these reasons, the implementation of the MyStoryPlayer client has been realized by using Flash/ActionScript development tool kit since the needed flexibility and control have been verified.

The MyStoryPlayer Client tool has been designed by using object-oriented technology and it satisfies the above reported requirements. Its design is reported in Fig. 10, where you can see

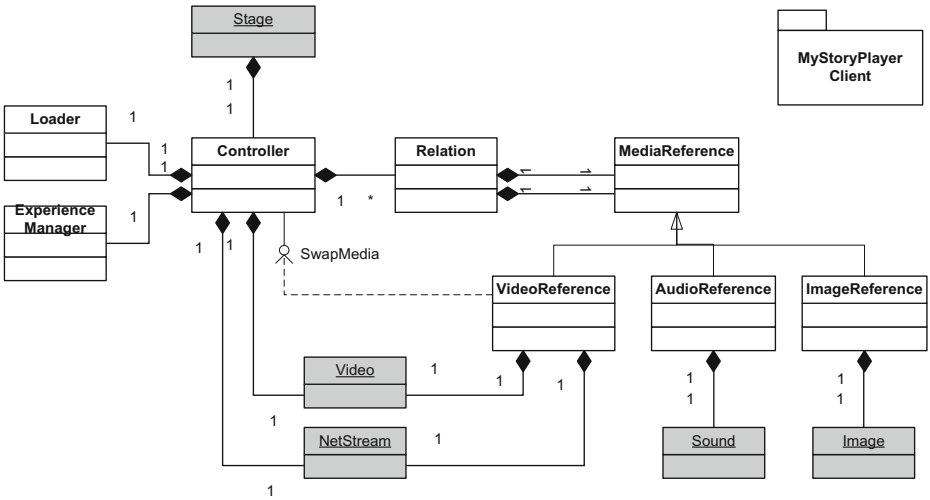


Fig. 10 The main classes of MyStoryPlayer client tool and their relationships. The classes reported in *grey* are those provided by the ActionScript library and framework

(i) the Loader to query, download and interpret the RDF triples of the involved relationships with the master media, the media relationships are instantiated as MediaReferences according to their model; (ii) the Experience Manager to collect user experiences as described in Section 4.1, model them as RDF, send them to server, download and play them according to user requests; and (iii) the Controller which manages the change of context (e.g., swap and back), the management of multiple audiovisual streams, the synchronization management, etc. Streams are received as progressive download flash video/audio streams. Images are directly associated with a time duration.

7.2 Inside controller and synchronizations

In order to explain better the problems related to the implementation of MyStoryPlayer Client, the activities of the Controller class have to be better analyzed. The Controller class manages (i) the initialization of the different instantiated media player and context by activating the Loader of RDF relationships, (ii) the creation/allocation of media Relation that creates specific areas on the user interface of the MyStoryPlayer and communicates with the Loader to obtain information about each annotation (associated to relations) in order to display them in the description box under the main media player area, (iii) the activation of Experience Manager.

Once the reproduction starts, the singleton of Controller class manages *onLoop()* function by which the event management is performed as a periodic service. Among the events to be managed by the Controller there are: user interactions on video controls (play, pause, seek, stop, volume control, etc.), context change (e.g., clicks to jump, swap and back), and events coming from the network streams management (e.g., buffer empty, buffer full, for each medium). The exploitation of these skills makes possible the management of multiple streams, relations and synchronizations among media through activities of preloading, buffering and seeking of videos on the basis of suitable seeking algorithms, which may depend on the events generated by the NetStream.

The main aim of MyStoryPlayer Client is to provide users with a good quality experience. In order to guarantee a suitable synchronization among media according to the expectations of the user who has established the relationships, it is important to minimize desynchronizations and to re-establish synchronizations among media when:

1. they are synchronized via *One2One* and Reciprocal Synchronization Relationships. They can lead to code the contemporaneous start of a set of videos or the start of additional video streams at a given time instant, while others are in execution.
2. the master medium execution is requested to perform a jump, for example when the user clicks on the timeline of the master video. The jump can be backward or forward, inside or outside the already buffered video stream. In presence of synchronized media, all of the synchronized media have to jump as well, while re-establishing the synchronizations in short time.
3. one of more video streams end their cumulated buffer, and they do not have frames to be played any longer. If this happens to the master, all other video streams should wait for the new buffering of the master to restart. When this happens to other video streams, large cumulative desynchronizations may be generated.
4. the context changes for the execution of *Exp* relationships the other media are stopped to live time at the exploded media segment to be played, and then return back to the master in the same time instant. This means that the other media can continue to the buffered while are stopped.
5. the context changes for a swap/back of media, intentionally performed by the user (clicking on a video stream on the right side of the MyStoryPlayer). In presence of new

context, a set of synchronized media may be available and all the streams have to restart and rapidly reach the synchronization again. The new condition can be anticipated by pre-buffering all the media of all possible scenarios which can be reached by a swap, thus saving the stack of buffered video. On the other hand, this solution is very expensive in terms of bandwidth and it cannot be applied.

The solutions in the state of the art are typically based on creating a muxed stream compounding multiple video streams to prepare precooked synchronizations to be streamed to the clients, in Section 2.3. This approach is viable for broadcasting applications (for example in MPEG-2 TS), and the client player can be quite simple, while the coded synchronizations and relationships can be changed in real time with some difficulties. A surrogate of changing stream/channel can be implemented by jumping from sub-streams belonging to the same muxed multiple streams, similarly to what happen to camera selection of F1 Gran Premium on satellite broadcasting, where you may have transported streams with multiple program streams.

In order to have a more dynamic solution, media streams have to reach the client player as independent streams. Multiple video streams could be provided by using RTSP server to some custom players; they could be kept synchronized according to the RTSP protocol forcing to establish specific client–server connection for each stream. On the other hand, today the most widespread solutions for web-based video players are typically based on simpler and less expensive http progressive download (see for example Flash players, and HTML 5). In [24], the multiple video streams are received by a custom player. In this case, in order to keep videos synchronized with one another, the custom player has implemented strategies like skipping frames and jumping / seeking on the local stream. This approach has constrained the solution to receive the full stream of non compressed videos, so as to be able of selecting frames. This last solution brings forth the installation of a custom video player and does not optimize the exploitation of network, since the player needs in any case to get the full rate video stream. Thus is not suitable for low bandwidth conditions. The aim of the proposed solution is to guarantee a suitable synchronization among media according to user expectations with no need of creating any inter-stream synchronization on the server side, nor showing video streams that could be even served by multiple independent HTTP servers, and thus minimizing desynchronizations.

The early implementation of the MyStoryPlayer Client (called in the following Solution 0) was based on starting the HTTP video streams when needed as in [10]. This approach leads to desynchronizations especially in the event of low bandwidth. In order to reduce these problems, the size of stream buffer to be cumulated before the video starts can be increased. On the other hand, when it comes to long videos to be kept synchronized, such as lessons, the probability of cumulating relevant delays is very high, thus resulting in increasing delays and variance of the desynchronizations among the streams, especially with low bandwidth (see Section 7.3).

In order to provide higher quality experience, several new MyStoryPlayer Client solutions have been created and tested. All the new proposed solutions are based on HTTP progressive download to enable the usage of multiple servers with no need of compounding muxed streams of videos, and using streaming servers with RTSP solution, as described above. The new solutions include algorithms combining a set of techniques: (i) performing a client side seeking into the stream buffer already loaded by the client (thus performing a sort of frame skip on the client side, via SeekClientTime), (ii) requesting a seeking of the video stream to the MyStoryPlayer Server (SeekServerTime) on server side (using xmoov server), thus reducing the network workload to send frames that are not necessary, (iii) adjusting delay in actuating the seeking on the client side (DelayToSeekClient), (iv) adjusting delay in obtaining a frame seeking from the server side and seeing it actuated on the client (DelayToSeekServer), this approach avoids the workload of the network with frames that are not played, (v) adjusting

delay in starting a video at a certain time instant, for example in the case of One2One relation (DelayToStart). The first step is to detect if the position to be sought is inside the already buffered segment of the video, this is performed by using Slave.BufferIsIn(MasterTime+ DelayToSeekClient).

The mentioned DelayToSeekClient, DelayToSeekServer, and DelayToStart are continuously estimated to create an adaptive adjustment solution. This approach is needed since the available network throughput is not stable and may change from stream to stream. The adaptive estimation model is based on the running average of the last three measured values for the same video stream. The measure of those delays is possible since the status of the streams is traced. More sophisticated algorithms are difficult to be implemented since the *onloop()* function has to run all action in 200 ms for all videos. The DelayToStart can be taken into account to anticipate the video buffering, when *One2One* relationships are approaching. Yet, in the event of low bandwidth it can create destructive effects to the current rendering. The HTTP protocol and ActionScript have limited capabilities in controlling the video buffer on client side when compressed video code is used, as in this case. Moreover, the seeking has to be avoided if the desynchronization is lower than the acceptable threshold, thus avoiding useless oscillating jumps in the video play. This limit is related to the resolution of the *OnLoop()* function of the Control singleton in the player. As to Flash player, we have a cycle every 200 ms, which could be an acceptable desynchronization error for most video lessons.

A simplified algorithm is reported in the following pseudocode:

```
DeSync=abs(MasterTime – Slave.CurrentTime);

If ( DeSync>MinDeSync) then

        If (Slave.BufferIsIn(MasterTime+DelayToSeekClient) ) then
            Slave.SeekClientTime(MasterTime+DelayToSeekClient);
        else Slave.SeekServerTime(MasterTime+DelayToSeekServer);

Endif
```

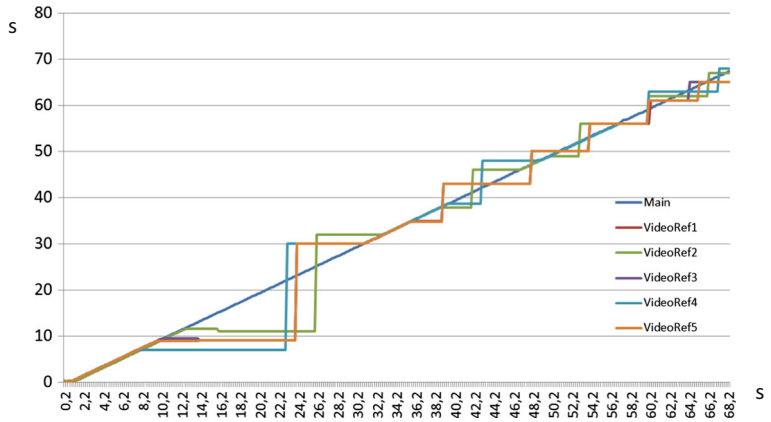


Fig. 11 Trends of video time codes into the first 68 s in executing TC1 with Solution 2

Table 1 Values estimated for Solution 2 on TC1, in the first 70 s

TC1	Mean error	Variance
1,600 Kbps	2.13	39.18
2,400 Kbps	1.12	16.55
3,200 Kbps	0.56	11.60
4,000 Kbps	0.42	2.33

In order to assess the system behavior and the improvement three different solutions have been taken into account. Solution 0, which has been early proposed, Solution 1: implementing buffering and taking into account the above described algorithm addressing (i) SeekClientTime(); and Solution 2, which has been realized by improving Solution 1 with the management of (ii) SeekServerTime(), (iii) DelayToSeekClient, (iv) DelayToSeekServer, and (v) DelayToStart.

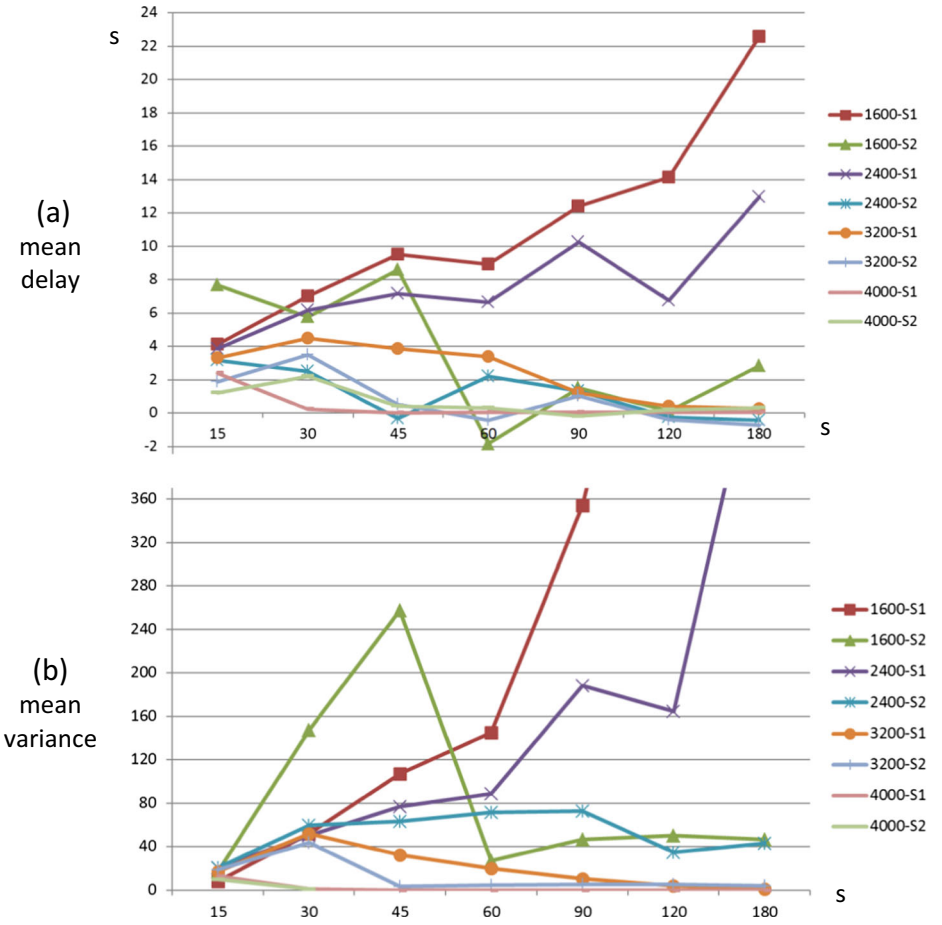


Fig. 12 Results of TC1 for Solutions 1 and 2 are reported for bandwidth ranging from 1,600 to 4,000 Kbps with respect to time in seconds (from 15 to 180, i.e., 3 min): **a** mean delay in s, **b** mean variance. In (**a**), negative values describe conditions where slave videos following the master have anticipated their master position

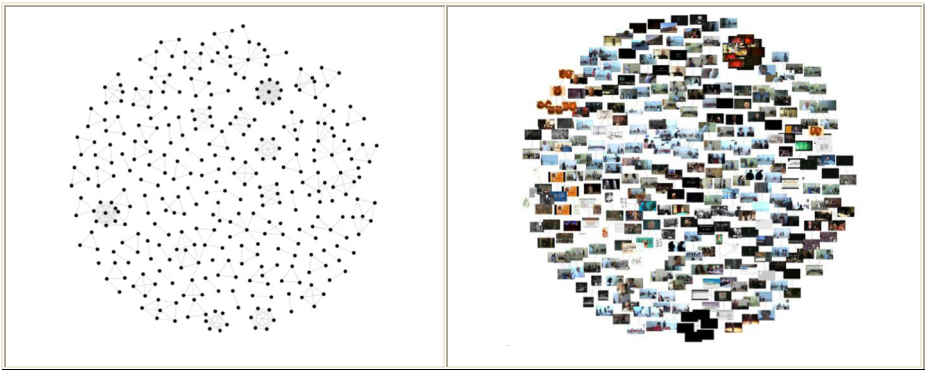


Fig. 13 Relationships among audiovisual elements in ECLAP. This analysis can be navigated at the URL: http://www.eclap.eu/d3/graph.html and http://www.eclap.eu/d3/graph2.html

7.3 Synchronization performance of MyStoryPlayer

In general, the video buffering could improve the quality of synchronizations among multiple videos on the same player. On the other hand dealing with multiple video streams on a limited network bandwidth and setting up a long buffer (for example 10 s) would lead to wait for the player to start for many seconds, since not all the video streams fill the buffer at the same speed. However, the limit of bandwidth and the number of media under simultaneous execution are issues which could reduce the performances and increase the delays among media. For this reason, the buffer has been kept limited to 1 s in both cases, so as to keep this parameter stable. According to our experiments, the imposed value is a good compromise to get quality without forcing users to wait for longer time span.

It is difficult to provide measures of performances which cover all the possible cases and conditions. On such grounds two main cases are taken as reference to show the obtained performance and improvements passing from Solution 0 to Solution 2 and using videos of 384×288 pixels, 25 fps, in H264 compression, with audio and an overall bit rate for each single video of 407 Kbps (audio plus video):

- **TC1 (Test Case 1)**: a master video synchronized with five videos (One2One relationships) since time 0. This resulted in six synchronized videos that should keep (or reach) the synchronization in different network bandwidth conditions and when the user performs jump forward and backward, swaps to one of the related video, etc. This TC1 resulted in a demand of network bandwidth of 2,442 kbps, and can be played on ECLAP with MyStoryPlayer by

Table 2 Metrics describing the usage of MyStoryPlayer model among media into the RDF database

Number of defined relationships	1,086
Number of media involved in some relationships	562
Only as a recipient of a relationship	46
Both as target and recipient of some relationships	272
Average number of relationships per media	1.93
Maximum number of relationships per media	11
Number of additional simple textual annotations on media segments	135

clicking on: http://www.eclap.eu/portal/?q=msp&axoid=urn%3Aaxmedis%3A00000%3Aobj%3Af3783c88-62bf-4960-ba1c-18da63e783b6&axMd=1&axHd=0 A recorded video demonstration about TC1 is also accessible at: http://www.eclap.eu/177806

- **TC2 (Test Case 2)**: a master video with other five One2One relationships of synchronization with other videos starting at 20 s of time. This resulted in six synchronized videos that should wait for 20 s and then start keeping the synchronization in different network bandwidth conditions and when the user performs jump forward and backward, swaps to one of the related video, etc. This TC2 resulted in a demand of network bandwidth of 407 kbps for the first 20 s, and then of 2,442 kbps, and can be played on ECLAP portal with MyStoryPlayer by clicking on: http://www.eclap.eu/portal/?q=msp&axoid=urn%3Aaxmedis%3A00000%3Aobj%3Ad5b311d0-bda8-4f42-81ab-a8f00b059f53&axMd=1&axHd=0

The Solution 0 against TC1 provided an average delay of 11.2 s after 60 s of play with a variance of 32.5 (at 2,400 Kbps, i.e. the bandwidth was constant). These values have been estimated by performing 10 tests/measures in the same conditions, by using tools to limit the network bandwidth. At 4,000 Kbps the conditions were slightly better providing an average delay of 7.7 s and a variance of 15.2. Similar results are obtained with respect to TC2 where the following measures have been taken: at 2,400 Kbps after 60 s, an average of 9.9 s of delay with a variance of 30; and at 4,000 Kbps after 60 s, an average of 7.6 s with a variance of 12.8. These conditions are obviously not acceptable since they bring in very large delay after 1 h of lesson rendering.

In order to explain the behavior of the best identified algorithm, in Fig. 11, the behavior of Solution 2 against TC1 in the first 68 s is reported for the limit case of 2,400 Kbps, while the six videos need a bandwidth of 2,442 Kbps (thus it is a sort of limit case). It can be noted that: (i) in the first time instants also the master/main video did not immediately start, (ii) the other videos waited for its start and the whole set had a problem around 8–9 s. The problem provoked a large desynchronization of 3 videos out of 6, then according to the proposed solution, there is a progressive error reduction due to the adaptive correction of the seeking delays, while at the same time some videos are perfectly following the master/main video.

In other execution of TC1 with lower bandwidth, it may happen that the master video runs out of buffering, thus forcing all the other videos to stop their play and continue their buffering. Those cases are less critical for the execution since the general execution time is delayed and the other videos have more time to buffer than in the case presented in Fig. 11. Please note that the proposed solution may lead to overestimate the DelayToSeekServer, thus causing corrective anticipative values.

In order to complete the view, Table 1 reports the mean error and variance for the first 70 s of Solution 2. The estimation has been obtained by executing ten times of TC1 with different values of network bandwidth.

In the several experiments, the noticed ranges for the corrective values adaptively estimated (as running averages of past delays) have been measured to be: DelayToSeekClient [0.2–0.4 s], DelayToSeekServer [1.2–6.0 s], and DelayToStart [1.2–15.0 s]. As described before, their current value is adaptively estimated at run time on the basis of the previous conditions. The DelayToSeekClient depends on the hardware hosting the MyStoryPlayer Client tool, while the others also depend on network bandwidth. In the event of jump forward and backward by the user, the seeks can be inside the loaded buffer or not, thus bringing in delays as DelayToSeekClient or DelayToSeekServer, respectively. These user provoked delays cannot be anticipated, but they can be corrected as unpredicted stops for re-buffering.

In order to compare the above mentioned solutions 1 and 2, a large set of experiments has been carried out to measure delays among different conditions. To this end, in Fig. 12, the results of TC1 for solutions 1 and 2 are reported, with a bandwidth ranging from 1,600 to 4,000 Kbps. Please note that Solution 1 cannot limit the average delay with low network bandwidth, while Solution 2 has strongly improved the performance and allowed to keep under control the mean value of delay, even with low network bandwidth. For example, solution 2 obtained after 3 min at 1,600 kbps, a mean delay of about 3 s with a variance of 42. Slightly better results are obtained for TC2 where for the first 20 s the master video has the time to buffer. As we can see from Fig. 12, in Solution 1 the average delay tends to augment in the event of insufficient bandwidth, if compared with the one needed for the current streams. In those cases, the values of variance augment exponentially and lead to instability as well.

8 Some use data of MyStoryPlayer within ECLAP

The MyStoryPlayer tool which has been presented in this article refers to its integration with ECLAP. ECLAP (European Collected Library of Performing Art) is a social network among performing art professionals with more than 170,000 content elements for more than 1.1 million items. In ECLAP, the MyStoryPlayer is mainly exploited to: compare performing art performances, present master classes and workshops in the performing arts field, with most of the related content being provided by CTFR (Centro Teatrale Franca Rame and Literature Nobel Prize Dario Fo) and CAT (Centro Teatrale Ateneo University of Rome), plus content coming from other 25 partners from all over Europe.

What has been developed is a system able to keep trace of the actions performed by users on the player, like swap, back, seek to a specific point of the timeline, reload of new media after a query on the system, and so on. This has been performed to monitor what users do with MyStoryPlayer, in order to understand better which direction the development of the tool had to follow, according to the user behavior with MyStoryPlayer. In the last months the outcome was an average of about ten clicks made by users along their navigation on the player, which is a good result, when considering that this tool is completely new, with new features, and users need time to get used to all the proposed features.

At present, there are more than 1.000 relationships involving more than 500 media (see Fig. 13 and Table 2 for other measures). Accesses are mainly performed by students for master classes and to play the relationships established by teachers and researchers in that sector.

As to the analysis of the structure of relationships created by users, it is very interesting to see the produced graphs of relationships and their dimension (see Fig. 13, where nodes are represented by media and edges by the relationships among them). In the full set of ECLAP audiovisuals, there are some connected and unconnected elements. In the general set, it has been possible to identify a number of separate groups of media. Inside each group, a number of relationships has been defined, for example putting in relationship the video sequences, the video synchronizations with several additions performed to provide examples and similarity annotations and audiovisual comments. In Table 1, some measures have been reported to provide the reader with a general idea about the relationships established among the media. Moreover, among the largest set of media relationships there are: 22 Media related by 26 Relations (single and reciprocal), described as (22M, 26R), other relevant examples are: (24M, 34R), (14M, 67R), (10M, 90R), etc. There are also very simple examples (1M, 1R), where the same medium has been annotated with itself (for example, two different segments of the same video).

9 Conclusions

The work presented in this article addressed the issues of education and training cases where multi-camera views are needed: performing arts and news, medical surgical actions, sport actions, instruments playing, speech training, etc. In most cases, users (both teachers and students) need to interact to establish among audiovisual segments relationships and annotations with the purpose of: comparing actions, gesture and posture; explaining actions; providing alternatives, etc. In order to keep the desynchronization problem among audiovisuals limited, most of the state of the art solutions are based on custom players and/or specific applications forcing to create custom streams from server side; this brings about restrictions on the user activity as to any dynamical establishing of relationships and access to the lessons via web. In this paper, MyStoryPlayer/ECLAP solution has been presented, providing: (i) a semantic model to formalize the relationships and play among audiovisuals determining synchronizations (One2One, Explosion, Reciprocal, and Sequential), (ii) a model and modality to save and share user experiences in navigating among related audiovisuals, (iii) solution to reduce the production of relationships, (iv) the architecture and the design of the whole system including the interaction model, and finally (v) the solution and algorithm to keep the desynchronizations among media limited, especially with low network bandwidth. The resulting solution includes a uniform semantic model, a corresponding semantic database for the knowledge, a distribution server for semantic knowledge and media, and the MyStoryPlayer Client for web applications. The proposed solution has been validated and it is at present in use within ECLAP (European Collected Library of Performing Arts, http://www.eclap.eu) to access and comment performing arts training content. The paper has reported validation results, as well, about performance assessment and tuning the media synchronization in critical cases. The validation test has demonstrated that the proposed solution is suitable for rendering multiple synchronized media via web in the event of low bandwidth, thus providing the user with the possibility of performing jumps (backward and forward), swap and back among media, etc. In ECLAP, users may navigate in the audiovisual relationships, while creating and sharing experience paths; at present several media relationships have been created and they are accessible to users and students of the institutions associated with ECLAP.

Acknowledgments The authors would like to express their thanks to Dario Fo, Franca Rame, Mariateresa Pizza of CTFR; Ferruccio Marotti, Raffaella Santucci of CTA UNIROMA, for their materials and support in tuning the solution with early test cases and trials. Sincere thanks to all the partners involved in ECLAP, and to the European Commission for funding ECLAP in the Theme CIP-ICT-PSP.2009.2.2, Grant Agreement No. 250481.

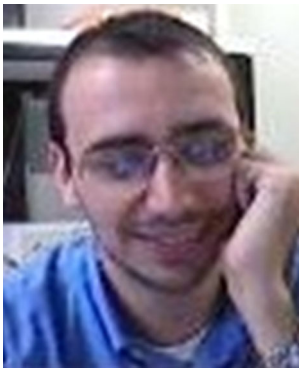
Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

1. AAF/MXF: Multimedia Exchange Format, http://www.ist-nuggets.tv/mxf
2. Allen E, Clarke E (1981) Design and synthesis of synchronization skeletons using branching-time temporal logic. Logic of Programs, Workshop. Spinger Verlag, London, UK, pp 52–71

3. AXMEDIS. Framework and Tools Specifications http://www.axmedis.org
4. Bellini P, Nesi P (2013) A Linked Open Data service for performing arts. Proc. of the ECLAP 2013 conference, 2nd International Conference on Information Technologies for Performing Arts, Media Access and Entertainment. Springer Verlag LNCS
5. Bellini P, Nesi P, Ortimini L, Rogai D, Vallotti A (2006) Model and usage of a core module for AXMEDIS/MPEG21 content manipulation tools. IEEE International Conference on Multimedia & Expo (ICME 2006), Pages 577–580 Toronto, Canada, 9–12 July, 2006.
6. Bellini P, Nesi P, Rogai D (2007) Exploiting MPEG-21 File Format for Cross Media Content. 13th International Conference on Distributed Multimedia Systems (DMS), San Francisco Bay, USA, 6–8 September 2007
7. Bellini P, Nesi P, Rogai D (2009) Expressing and organizing real time specification patterns via temporal logics. J Syst Softw, Elsevier, pp 1–36, vol. 82, N.2
8. Bellini P, Bruno I, Nesi P (2011a) Exploiting intelligent content via AXMEDIS/MPEG-21 for modelling and distributing news. Int J Softw Eng Knowl Eng, World Scientific Publishing Company, Vol. 21, n.1, pp 3–32
9. Bellini P, Nesi P, Paolucci M, Serena M (2011b) Models and tools for content aggregation and audiovisual cross annotation synchronization. Proceeding of ISM '11 Proceedings of the 2011 I.E. International Symposium on Multimedia, Pages 210–215, ISBN: 978-0-7695-4589-9, doi:10.1109/ISM.2011.41, 5–7 December 2011, Dana Point (Cam USA)
10. Bellini P, Nesi P, Serena M (2011c) Mystoryplayer: Semantic Audio Visual Annotation And Navigation Tool, proc of the 17th international conference on Distributed Multimedia Systems, Convitto della Calza, Florence, Italy, 18–20 August 2011
11. Bellini P, Cenni D, Nesi P (2012) On the Effectiveness and Optimization of Information Retrieval for Cross Media Content. Proceeding of the KDIR 2012 is part of IC3K 2012, International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 4–7 October 2012, Barcelona, Spain
12. Blakowski G, Steinmetz R (1996) A media synchronization survey: reference model, specification, and case studies. IEEE J Sel Areas Commun 14(1):5,35
13. Boll S, Klas W (2001) ZYX-A multimedia document model for reuse and adaptation of multimedia content. IEEE Trans Knowl Data Eng 13:361–382. doi:10.1109/69.929895
14. Boronat F, Lloret J, Garcia M (2009) Multimedia group and inter-stream synchronization techniques: a comparative study. Inf Syst 34(1):108–131. doi:10.1016/j.is.2008.05.001, ISSN 0306–4379
15. Broekstra J, Kampman A, van Harmelen F (2002) Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. First International Semantic Web Conference (ISWC'02), Sardinia, Italy, June 9–12, 2002
16. Bulterman DCA, Hardman L (2005) Structured multimedia authoring. ACM Trans Multimed Comput Commun Appl 1(1):89–109
17. Bulterman DCA, Rutledge LW (2009) SMIL3.0—Interactive Multimedia for Web, Mobile Devices and DAISY Talking Books. Springer-Verlag. ISBN: 978-3-540-78546-0
18. Burnett IS, Davis SJ, Drury GM (2005) MPEG-21 digital item declaration and identification—principles and compression. IEEE Trans Multimed 7(3):400–407
19. Chilamkurti N, Zeadally S, Soni R, Giambene G (2010) Wireless multimedia delivery over 802.11e with cross-layer optimization techniques. Multimed Tools Appl 47:189–205. doi:10.1007/s11042-009-0413-6
20. Dakss J, Agamanolis S, Chalom E, Bove Jr VM (1998) Hyperlinked video. Proc SPIE Multimed Syst Appl, v. 3528
21. Dublin Core Metadata Initiative, http://dublincore.org/
22. EDM, Europena (2010) Europeana Data Model Primer. Technical report. August 2010. Retrieved April 30, 2011, from http://version1.europeana.eu/web/europeana-project/technicaldocuments/ http://pro.europeana.eu/documents/900548/bb6b51df-ad11-4a78-8d8a-44cc41810f22
23. Gaggi O, Danese L (2011) A SMIL player for any web browser, Proc. Of Distributed Multimedia Systems, DMS2011, Florence, Italy, pp 114–119
24. Gao B, Jansen J, Cesar P, Bulterman DCA (2011) Accurate and low-delay seeking within and across mash-ups of highly-compressed videos. In: Proceedings of the 21st international workshop on network and operating systems support for digital audio and video (NOSSDAV '11). ACM, New York, pp 105–110. doi:10.1145/1989240.1989266
25. Hausenblas M (2008) Non-linear interactive media productions. Multimedia Systems 14(6):405–413. doi:10.1007/s00530-008-0131-3
26. ITU-T Rec. H.761, Nested Context Language (NCL) and Ginga-NCL for IPTV Services, Geneva, Apr. 2009. Available at http://www.itu.int/rec/T-REC-H.761. Last accessed on 25th June 2012
27. Kahan J, Koivunen M, Prud'Hommeaux E, Swick R, Annotea: An Open RDF Infrastructure for Shared Web Annotations. In: Proc. of the WWW10 International Conference, Hong Kong, May 2001 http://www10.org/cdrom/papers/488/index.html http://www.w3.org/2001/Annotea/

28. Klamma R, Spaniol M, Renzel D (2006) Virtual entrepreneurship lab 2.0: sharing entrepreneurial knowledge by non-linear story-telling. RWTH Aachen University, Germany. J Univ Knowl Manag, Springer, 1(3):174–198
29. Koivunen M, Swick R, Prud'hommeaux E "Annotea Shared Bookmarks", 2003, In Proc. of KCAP 2003, http://www.w3.org/2001/Annotea/Papers/KCAP03/annoteabm.html
30. Kosovic D, Schroeter R, Hunter J (2004) FilmEd: Collaborative Video Annotation, Indexing and discussion tool over Broadband networks. Proceedings of the 10th International Multimedia Modelling, Page: 346
31. Layada N, Sbry-Ismail L, Roisin C (2002) Dealing with Uncertain Durations in Synchronized Multimedia Presentations. Multimed Tools Appl. Kluwer Academic Publishers, vol. 18, pp. 213–231. doi:10.1023/A:1019944800320
32. Lee I, Park J (2010) A scalable and adaptive video streaming framework over multiple paths. Multimed Tools Appl 47:207–224. doi:10.1007/s11042-009-0414-5
33. Ligne de Temps. http://www.iri.centrepompidou.fr/outils/lignes-de-temps/
34. Lombardo V, Damiano R (2011) Semantic annotation of narrative media objects. Multimed Tools Appl 59(2):407–439
35. Mayer-Patel K, Gotz D (2007) Scalable, adaptive streaming for nonlinear media. IEEE Multimed 14(3):68–83. doi:10.1109/MMUL.2007.63
36. Meixner B, Hoffmann J (2012) Intelligent download and cache management for interactive non-linear video. Multimedia Tools and Applications
37. Meixner B, Kosch H (2012) Interactive non-linear video: definition and XML structure. In: Proceedings of the 2012 ACM symposium on Document engineering (DocEng '12). ACM, New York, pp 49–58. doi:10.1145/2361354.2361367
38. MPEG-21 DIS: http://mpeg.chiariglione.org/technologies/mpeg-21/mp21-dis/index.htm
39. MPEG-7: http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm
40. Neuschmied H (2007) MPEG-7 Video Annotation Tool Media Analyze—Pre-processing Tool MPEG-7 Video Annotation Tool. pp 5–7
41. OpenAnnotation of W3C, http://www.openannotation.org/, http://www.openannotation.org/spec/core/
42. OverlayTV: http://www.overlay.tv/
43. Pereira F, Ebrahimi T (eds) (2002) The MPEG-4 book. IMSC Press, Prentice Hall
44. RDF semantic Web Standard, http://www.w3.org/RDF/
45. Schroeter R, Hunter J, Kosovic D (2003) Vannotea—A Collaborative Video Indexing, Annotation and Discussion System for Broadband Networks. Proc of Knowledge Markup and Semantic Annotation Workshop, K-CAP 2003, USA
46. SCORM (2003) Advanced Distributed Learning. The Sharable Content Object Reference Model (SCORM) - Version 1.3 - WD
47. Sesame: http://www.openrdf.org/doc/sesame/users/index.html
48. Shen E, Lieberman H, Davenport G (2009) What's next?: emergent storytelling from video collection. In: CHI '09: Proceedings of the 27th international conference on Human factors in computing systems, pages 809. 818, ACM, New York
49. Shipman F, Girgensohn A, Wilcox L (2008) Authoring, viewing, and generating hypervideo: an overview of Hyper-Hitchcock. ACM Trans Multimed Commun Appl 5(2), Article 15, 19 pages
50. Smith JR, Lugeon B (2000) A Visual Annotation Tool for Multimedia Content Description, Proc. SPIE Photonics East, Internet Multimedia Management Systems
51. SPARQL. Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/
52. Steves MP, Ranganathan M, Morse EL (2000) SMAT: Synchronous Multimedia and Annotation Tool. Proceedings of the International Conference on System Science, September, 2000
53. Tsai MF, Shieh CK, Ke CH, Deng DJ (2010) Sub-packet forward error correction mechanism for video streaming over wireless networks. Multimed Tools Appl 47:49–69. doi:10.1007/s11042-009-0406-5
54. Ursu Marian F, Thomas M, Kegel I, Williams D, Tuomola M, Lindstedt I, Wright T, Leurdijk A, Zsombori V, Sussner J, Myrestam U, Hall N (2008) Interactive TV narratives: opportunities, progress, and challenges. ACM Trans Multimed Comput Commun Appl 4(4), Article 25
55. Venkat Rangan P, Ramanathan S, Kaeppner T (1995) Performance of inter-media synchronization in distributed and heterogeneous multimedia systems. Comput Netw ISDN Syst 27(4):549–565. doi:10.1016/0169-7552(93)E0112-R
56. YouTube Creating or editing annotations. Website: https://support.google.com/youtube/?hl=en-GB#topic=2676319
57. Zhai G, Fox G, Pierce M, Wu W, Bulut H (2005) eSports: Collaborative an Synchronous Video Annotation System in Grid Computing Environment. Proc. of Seventh IEEE International Symposium on Multimedia, 12–14 December, CA, USA



Pierfrancesco Bellini is a contract Professor at the University of Florence, Department of Systems and Informatics. His research interests include object-oriented technology, real-time systems, formal languages, computer music. Bellini received a PhD in electronic and informatics engineering from the University of Florence, and has worked on projects funded by the European Commission such as: ECLAP, AXMEDIS, MOODS, WEDELMUSIC, IMUTUS, MUSICNETWORK, VARIAZIONI and many others. He has been co-editor of MPEG SMR.

Paolo Nesi is a full professor at the University of Florence, Department of Information Engineering, chief of the DISIT, Distributed Data Intelligent Technology lab and research group. His research interests include multimedia modeling, knowledge representation, parallel and distributed systems, content protection, DRM, P2P, physical models semantic computing, real-time systems, formal languages, and computer music. He has been the general Chair of DMS, IEEE ICSM, IEEE ICECCS, WEDELMUSIC, AXMEDIS, ECLAP, international conferences and program chair of several others. He has been the coordinator of several R&D multipartner international R&D projects of the European Commission such as ECLAP, AXMEDIS, MOODS, WEDELMUSIC, MUSICNETWORK and he has been involved in many other projects. He has been co-editor of MPEG SMR. Contact Paolo Nesi at paolo.nesi@unifi.it.

Marco Serena is a research contractor at the University of Florence. He is an Engineer and now a Ph.D. candidate in the 2013. His research interests include media computing, semantic computing, grid computing, image processing. He has worked on ECLAP European Commission project.